ADIS - A robust pursuit algorithm for probabilistic, constrained and non-square blind source separation with application to fMRI

Gautam Pendse*[1] David Borsook[1]
Lino Becerra[1]

[1] Imaging and Analysis Group (IMAG), Harvard Medical School

Feb 27, 2009

*To whom correspondence should be addressed. e-mail: gpendse@mclean.harvard.edu

Abstract

In this article, we develop an algorithm for probabilistic and constrained projection pursuit. Our algorithm called ADIS (automated decomposition into sources) accepts arbitrary non-linear contrast functions and constraints from the user and performs *non-square* blind source separation (BSS). In the first stage, we estimate the latent dimensionality using a combination of bootstrap and cross validation techniques. In the second stage, we apply our state-of-the-art optimization algorithm to perform BSS. We validate the latent dimensionality estimation procedure via simulations on sources with different kurtosis excess properties. Our optimization algorithm is benchmarked via standard benchmarks from GAMS performance library. We develop two different algorithmic frameworks for improving the quality of local solution for BSS. Our algorithm also outputs extensive convergence diagnostics that validate the convergence to an optimal solution for each extracted component. The quality of extracted sources from ADIS is compared to other well known algorithms such as Fixed Point ICA (FPICA), efficient Fast ICA (EFICA), Joint Approximate Diagonalization (JADE) and others using the ICALAB toolbox for algorithm comparison. In several cases, ADIS outperforms these algorithms. Finally we apply our algorithm to a standard functional MRI data-set as a case study.

1 Introduction

The Generalized Linear Model (GLM) is a popular tool for analyzing functional MRI (fMRI) data. GLM analysis proceeds on a voxel by voxel basis using the same design matrix. One of the difficulties associated with GLM analysis is the construction of an appropriate design matrix. Unmodeled regressors that modulate the fMRI signal in addition to the EVs but are not a part of the design matrix will invalidate the analysis and the associated inferences. Further, these unmodeled regressors might be different in different brain regions and a voxel by voxel analysis with a common design matrix may not be appropriate. These considerations imply that model based analyses make very strong assumptions which are very likely violated in a real fMRI dataset.

Model free analysis on the other hand does not need any postulations as to the shape of the expected response. One such technique that has become popular in recent years, particularly for application to fMRI is Independent Component Analysis (ICA) [9]. See [25] for a survey on ICA. Popular software packages such as FSL ([35]) come with ICA software for doing non-square ICA via automatic latent dimensionality estimation also known as Probabilistic Independent Component Analysis (PICA) [2]. Essentially these techniques consists of a data reduction step using PCA followed by the application of standard ICA algorithms. In the signal processing community, many well established algorithms exist for doing ICA, the most popular ones being efficient FastICA [27], Fixed Point ICA (FPICA) [26], Joint Approximate Diagonalization of Cumulant Matrices (JADE) [6], Extended Robust ICA (ERICA) [13] and unbiased ICA (UNICA) [14]. In this article, we will question the validity of solution produced by current ICA codes. Are these solutions really optimal? Can we afford to pay the price for using non-optimal solutions?

One of the challenges involved in applying ICA to real data is the confidence in the quality of

estimated solution. This problem has been recognized before. For example the software package ICASSO [21] uses bootstrapping simulations to run an ICA algorithm multiple times and then clusters the estimated sources to assess reliability. ICASSO takes into account the "algorithmic" variability and the variability in the original data induced due to sampling, but since it assumes square, noise free mixing it ignores the estimation errors induced due to noisy source mixing. The presence of non-square mixing in real data also introduces additional variability due to the unknown latent dimensionality of the sources.

While a bootstrapping strategy can always be used to test the "sensitivity" of estimated BSS solution from any algorithm, it is critical to have a reliable and verifiable optimization solver to solve the non-convex BSS problem in the first place.

Currently existing ICA codes perform optimization using techniques that have formulas for updating the unknown variables using a Newton step, gradient descent, natural gradient [3], [1] [18] or similar strategies. In addition they also potentially have a number of heuristic rules for updating the various "learning" parameters in these algorithms. No convergence diagnostics are used to check the optimality of estimated solution. To the best of our knowledge, such optimization codes are not benchmarked using standard optimization test cases. Such ad-hoc strategies along with non-verified optimality may severely affect the quality of solution produced by these algorithms and potentially impact the practical conclusions drawn from incorrect results. The popular FastICA algorithm [27] uses an approximate Newton iteration where the approximate Hessian simplification reduces it to a gradient descent algorithm with a fixed step size. However there is no reason to use these approximations. One can use state of the art optimization software to compute near exact step sizes with locally varying Hessian approximations. In fact, it has been shown [39] that these exact step size search significantly increases the estimation efficiency and robustness to initialization in comparison to the fixed update rules of FastICA. The optimization core in our algorithm ADIS efficiently handles local non-convexity as well as allows for infeasible steps, i.e., steps that violate the constraints leading to a more fuller exploration of parameter space and increasing the likelihood of converging to a global optimum.

Another issue is the modification of the default contrast function in ICA (e.g. Negentropy) to do other types of source extraction. It is also desirable to be able to add additional equality and/or inequality constraints to BSS estimation depending on the application at hand. These issues cannot be addressed using current ICA software. In this paper, we develop our algorithm ADIS and validate its various components. Finally, we compare ADIS to currently existing ICA codes on many standard benchmark datasets from ICALAB.

Our algorithm ADIS (section 2, 3):

1. Uses a state-of-the-art optimization algorithm at its core (inspired by LANCELOT software [12]) (section 4, 14)
2. Uses a bootstrap simulation/cross-validation based approach for latent dimensionality estimation (in case of non-square BSS) (section 6)

3. Enables the user to use arbitrary contrast functions and nonlinear constraints for BSS
4. Produces "good quality" local solutions using a special multistage framework for BSS (section 3)
5. Produces extensive convergence diagnostics for each extracted component to validate the "optimality" of the extracted source (section 4)

We perform validation of each component of ADIS as follows:

1. Validation of the latent dimensionality estimation procedure using simulations on sources with different statistical properties (section 6)
2. Validation of our optimization core using standard benchmarks from the GAMS performance library (`http://www.gamsworld.org/performance`, [15]) (section 15)
3. We then use the "Negentropy" contrast function as a special case and compare the results of ADIS in terms of separation quality and robustness to other well known algorithms such as efficient FastICA, FPICA, JADE and others using the ICALAB toolbox [8], [7] for BSS algorithm comparison. (section 7)
4. Finally we apply ADIS to real fMRI data as a case study (section 8)

2 Probabilistic Projection Pursuit

Projection Pursuit is a standard statistical technique for data analysis [17],[16], [22]. In this article we generalize projection pursuit in a probabilistic framework similar to the one proposed by Beckmann et. al. ([2]). We consider data generation at n points via a noisy mixing process as follows:

$$x = \mu + A\,s + \eta,\ i = 1, 2, \ldots, n \tag{1}$$

where $x \in \mathbf{R}^p$, $s \in \mathbf{R}^q$ and $A \in \mathbf{R}^{p \times q}$, $\eta \sim \mathbf{N}(0, \sigma^2 V)$. We assume that $p > q$ to achieve a compact representation of the observed data.

The problem is to estimate automatically q, A, s and μ given observations $x_i, i = 1 \ldots n$. This problem is also called a blind source separation (BSS) problem since q, A and s are all unknown. The inclusion of noise term η makes the problem into a probabilistic one.

First consider the case when $V_i = I_p$ for all i. Section 5.1 shows how to handle the case $V_i \neq I_p, \forall i$.

If $1_p = [1, 1, \ldots, 1]^T \in \mathbf{R}^p$ be a vector of ones. Then

$$1_p^T x = 1_p^T \mu + 1_p^T A\, s + 1_p^T \eta, i = 1, 2, \ldots, n \tag{2}$$

Let

$$\bar{D} = D - 1_p 1_p^T D, \text{ where } D = x, A, \mu, \eta \tag{3}$$

Then we can write

$$\bar{x} = \bar{\mu} + \bar{A}s + \bar{\eta} \tag{4}$$

where $\bar{\eta} \sim \mathbf{N}\left(0, \sigma^2(I_p - 1_p 1_p^T/p)\right)$ Since the scaling of A and s is arbitrary we assume without loss of generality that

$$\mathrm{E}(s) = 0 \text{ and } \mathrm{Cov}(s) = \mathbf{I}_q \tag{5}$$

No other assumptions are made about the joint source density $p(s)$ other than the ones in 5. From equations 4 and 5:

$$\bar{\mu} = E(\bar{x}) \tag{6}$$

$$E[(\bar{x} - \bar{\mu})(\bar{x} - \bar{\mu})^T] = \bar{A}\bar{A}^T + \sigma^2(I_p - 1_p 1_p^T/p) \tag{7}$$

Using the law of large numbers (LLN) we can approximate the covariance matrix as:

$$E[(\bar{x} - \bar{\mu})(\bar{x} - \bar{\mu})^T] \approx \frac{1}{n}\sum_{i=1}^{n}(\bar{x}_i - \bar{\mu})(\bar{x}_i - \bar{\mu})^T = U\Sigma U^T \tag{8}$$

In 8, $\Sigma = diag(\lambda_k)$ with $\lambda_k, k = 1, \ldots, p$ the singular values and U a matrix containing the corresponding singular vectors of the covariance matrix. The estimate of μ is easily obtained from 6.

$$\hat{\bar{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\bar{x}_i \tag{9}$$

Without knowing the true source densities $p(s)$ it is not possible to estimate the maximum likelihood estimates of $\bar{A}$ and σ^2. However one can find estimates that try to satisfy the second order condition 7 as closely as possible by minimizing the Frobenius norm:

$$\hat{\bar{A}}, \hat{\sigma^2} = \underset{\bar{A},\sigma^2}{\operatorname{argmin}} ||\frac{1}{n}\sum_{i=1}^{n}(\bar{x}_i - \hat{\bar{\mu}})(\bar{x}_i - \hat{\bar{\mu}})^T - \bar{A}\bar{A}^T - \sigma^2(I_p - 1_p 1_p^T/p)||_F^2$$

It is easily shown that if U_q is a $p \times q$ submatrix of U containing the first q singular vectors corresponding to the q largest singular values and Σ_q is the $q \times q$ submatrix of Σ then the solution to 10 is given by:

$$\hat{\tilde{A}} = U_q(\Sigma_q - \hat{\sigma^2} I_q)^{1/2} Q^T \tag{10}$$

and

$$\hat{\sigma^2} = \frac{1}{p-q-1} \sum_{i=q+1}^{p-1} \lambda_i \tag{11}$$

where Q is an arbitrary $q \times q$ orthogonal matrix ($Q^T Q = I_q$). Given $\hat{\tilde{A}}$ and $\hat{\sigma^2}$, the least squares estimate of $\hat{s_i} \in \mathbf{R}^q$ are given by:

$$\hat{s_i} = (\hat{\tilde{A}}^T \hat{\tilde{A}})^{-1} \hat{\tilde{A}}^T (\bar{x}_i - \hat{\tilde{\mu}}) = Q(\Sigma_q - \hat{\sigma^2} I_q)^{-1/2} U_q^T (\bar{x}_i - \hat{\tilde{\mu}}) = Q\tilde{x}_i \tag{12}$$

where

$$\tilde{x}_i = (\Sigma_q - \hat{\sigma^2} I_q)^{-1/2} U_q^T (\bar{x}_i - \hat{\tilde{\mu}}) \tag{13}$$

3 Problems solved in Projection Pursuit

In Projection Pursuit (PP), we parameterize the orthogonal matrix Q as

$$Q = [w_1, w_2 \ldots, w_q]^T \tag{14}$$

where $w_i \in R^q$. The kth PP projection is defined as for each point i:

$$\hat{s_{ki}} = w_k^T \tilde{x}_i,\ i = 1 \ldots n \text{ and } k = 1 \ldots q \tag{15}$$

In vector form we can write the kth projection as:

$$\hat{s}^k = w_k^T \tilde{x} \tag{16}$$

where $\hat{s}^k = [\hat{s}_{k1}, \ldots, \hat{s}_{kn}]$ is a $1 \times n$ vector and $\tilde{x} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n]$ is a $q \times n$ matrix. We define an objective function $f(s^1, s^2, \ldots, s^q)$ and optimize for the projection vectors. Mathematically we solve the optimization problem:

$$[w_1^*, \ldots, w_q^*] = \underset{w_1,\ldots,w_q}{\operatorname{argmax}} f(w_1^T \tilde{x}, \ldots, w_q^T \tilde{x}) + b(w_1^T \tilde{x}, \ldots, w_q^T \tilde{x}; \Theta)$$

Here b is function accounting for supplementary information that we want to include in the optimization problem. Θ is all supplementary information of interest to the optimization problem. For example, in spatial problems Θ could be a spatial location of points and g could be a function accounting for spatial smoothness (such as a Markov random field). Many such problem specific functions can be proposed based on user objectives. There could also be additional user defined equality and inequality constraints, for example:

$$c_i(w_k^T \tilde{x}) = 0,\ i = 1 \dots m,\ , k = 1 \dots q \tag{17}$$

$$g_i(w_k^T \tilde{x}) \geq 0,\ i = 1 \dots L,\ , k = 1 \dots q \tag{18}$$

Thus in general, we get a constrained projection pursuit problem. Constraints 17 and 18 can be arbitrary non-linear constraints not necessarily parameterized by $w_k^T \tilde{x}$.

3.1 Separability

The optimization problem in 17 must involve a joint optimization of the vectors $w_1, \dots, w_q$ in general. In many important practical cases (such as for example when f is the joint negentropy index) the objective function f has a separable structure [26] such that

$$f(w_1^T \tilde{x}, w_2^T \tilde{x}, \dots, w_q^T \tilde{x}) = \sum_{k=1}^{q} h(w_k^T \tilde{x}) \tag{19}$$

for some function h, then the optimization can proceed sequentially where at each stage we solve

$$w_k^* = \ \arg\min_{w_k} h(w_k^T \tilde{x}) \tag{20}$$

At each stage w_k^* is a unit vector orthogonal to the previously calculated vectors i.e

$$w_k^{*T} w_l^* = \begin{cases} 0 & \text{when} \quad l < k \\ 1 & \text{if} \quad l = k \end{cases} \tag{21}$$

3.2 Multistage Optimization stragegy

In this section we describe a 2 or 3 stage strategy which we have found via experience to converge to good "local" solutions. It is well known that if an optimization problem that has multiple optima then the "local" solution found by an algorithm is strongly dependent on how the algorithm is initialized. For applications such as BSS, even though an algorithm finds a "local" solution as

indicated by convergence diagnostics, it may not be a "global" solution. In order to increase our chances of finding a solution that is global, we propose a random sampling strategy for initialilzation of an optimization algorithm first.

3.2.1 Stage 0: Search for good seed points

For concreteness, suppose $w_1^*, w_2^*, \ldots, w_{k-1}^*$ are the optimal solutions found previously and suppose we are tying to find w_k^*. Let $\tilde{W}$ be a $q \times (q-k+1)$ matrix that is the orthogonal complement of $[w_1^*, w_2^*, \ldots, w_{k-1}^*]$ as determined by say Gram-Schmidt orthogonalization. Then

$$\tilde{W}^T w_l^* = 0,\ l = 1, 2, \ldots, k-1 \tag{22}$$

1. Generate n_s vectors in $\mathbf{R}^q$ whose elements are drawn from a uniform distribution on $(-1, 1)$.
2. Standardize each vector to have unit norm to get the set of vectors $Z = [z_1, z_2, \ldots, z_{n_s}]$ where each $z_i \in \mathbf{R}^q$.

For each z_i we define seed points as

$$u_i = \tilde{W} z_i \tag{23}$$

It is easy to see that $u_i^T w_l^* = 0, l = 1, 2, \ldots, k-1$ and $u_i^T u_i = 1$. Thus u_i satisfies the constraints in 21. We then compute the objective function $h(w_k^T \tilde{x})$ at each of these points $u_i, i = 1, 2, \ldots, n_s$ and choose R points $t_1, \ldots, t_R$ that give the highest objective function values as candidate seed points for the next step.

3.2.2 Stage 1: Computing local optimum at R best points from Stage 1

In this stage, we compute the local solutions $w_k^{*1}, w_k^{*2}, \ldots, w_k^{*R}$ of the optimization problem 20 starting from $t_1, \ldots, t_R$ and choose the best local solution that has the highest function value.

$$w_k^* = \operatorname{argmax} h_k(w_k^{*i}), i = 1, 2, \ldots, R \tag{24}$$

ADIS uses $n_s = 1000$ and $R = 2$ as the defaults.

3.2.3 Stage 2: Joint optimization with initialization from Stage 2

After Stage 1 has been applied from $k = 1, \ldots, q$ we have an estimate of the solution vectors $w_k^*, k = 1, 2, \ldots, q$. In this stage we solve the joint optimization problem 19 subject to the single joint constraint:

$$\sum_{i=1}^{q} \sum_{j=i}^{q} (w_i^T w_j - \delta_{ij})^2 = 0 \tag{25}$$

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. We initialize the algorithm with the solution $w_k^*, k = 1, \ldots, q$ from Stage 1. The algorithm converges to a local joint solution only in a few iterations.

4 Optimization Algorithm

For flexible and powerful BSS algorithms, a primary requirement is a fast and robust optimization algorithm that can handle non-linear user defined constraints as well as handle non-convexity in the objective function or the constraints. Furthermore, extensive convergence diagnostics should be a standard output of the optimization process to ensure convergence to a local solution. The optimization core in ADIS (coded in MATLAB, `www.mathworks.com`) uses a modified augmented lagrangian algorithm (inspired by the implementation in LANCELOT package [10], [12]) to solve equality constrained problems generated in constrained non-convex BSS problems. Inequality constraints are handled by first transforming them to equality constraints via slack variables and solving the resulting bound constrained optimization problem. Some features of interest are as follows:

1. A Trust region based approach [30] is used to generate search directions at each step (for both equality constrained and inequality constrained problems).
2. For equality constraints only, the subproblems above are solved using a conjugate gradient approach (Newton-CG -Steihaug) [36] that is fast and accurate even for large problems and can handle both positive definite and indefinite hessian approximations. If both equality and inequality constraints are present then we solve the trust region problem with a non-linear gradient projection technique [5] followed by subspace optimization using Newton-CG-Steihaug. Our algorithm allows for infeasible iterates i.e., those that do not satisfy the problem constraints during optimization. This allows for a fuller exploration of parameter space and increases the likelihood of converging to a global optimum.
3. A symmetric rank 1 (SR1) quasi-Newton approximation to the hessian [11] is used which is known to generate good hessian approximations for both convex and non-convex problems. As suggested in [33] we do the update also on the rejected steps to gather curvature information about the function. We provide options for BFGS [4] especially for convex problems and an option for preconditioning the CG iterations. We also implement limited memory variants of SR1 and BFGS for large problems.
4. Our algorithm accepts vectorized constraints so that multiple constraints can be programmed simultaneously. Only gradient information is required. Hessian information is optional but not required. Optionally, it is easy to interface our code with INTLAB software package [34] for automatic differentiation in which case the user only codes the function and constraints and the gradients/hessians are generated automatically.

We tested the performance of our algorithm using standard optimization benchmarks from the GAMS performance benchmark problems (`http://www.gamsworld.org/performance`, [15]). The appendix shows some sample benchmarks as well as provides more technical details of the algorithm.

4.1 Convergence Diagnostics

It is *critical* to verify that the optimization algorithm has found a local solution by checking convergence diagnostics. These diagnostics help us determine if the Karush-Kuhn-Tucker (KKT) [33] necessary conditions for optimality have been satisfied or not. Profile plots are plots of a diagnostic measure versus iteration number. We propose the following checks for all BSS algorithms:

1. Convergence to a point satisfying necessary conditions for optimality can be accessed by looking at profile plots for
   - Objective function
   - Optimality error (such as "gradient of the lagrangian" for equality constraints or "KKT optimality checks" for general constraints)
   - Feasibility error (checking constraint satisfaction)
   - Lagrange multipliers
   - Other parameters in the algorithm (such as a barrier or penalty parameter)

   The user should at the very least check these diagnostic plots to make sure convergence is attained. If possible the second order sufficient conditions for optimality should also be verified at the solution point using Hessian information.
2. The algorithm used for optimization should **flag an error and stop running** in case convergence is not attained at any intermediate stage. This prevents the user from getting access to incorrect results.

These convergence diagnostics are a standard feature of ADIS. Any solution returned by ADIS is guaranteed to be optimal.

5 Statistics on estimated sources

In this section we develop equations that enable us to apply ADIS to a real data-set and make inferences from extracted sources.

Once $\hat{s}_i$ are estimated we can compute their variance using the GLM estimate

$$\hat{\text{Cov}}(\hat{s}_i) = (\hat{\bar{A}}^T \hat{\bar{A}})^{-1} \hat{\sigma_i^2} \tag{26}$$

where $\hat{\sigma_i^2}$ is the estimated variance at point i

$$\hat{\sigma_i^2} = \frac{(\bar{x}_i - \hat{\bar{\mu}} - \hat{\bar{A}}\hat{s}_i)^T (\bar{x}_i - \hat{\bar{\mu}} - \hat{\bar{A}}\hat{s}_i)}{p - q} \tag{27}$$

We can create maps of contrasts of interest using the above equations. We also estimate the relative variance (RV) contribution at point i using the component k as follows:

If $\hat{\bar{A}} = [a_1, a_2, \ldots, a_q]$ then

$$RV(k,i) = \frac{\text{Var}(a_k) \hat{s_{ki}}^2}{\sum_{k=1}^{q} \text{Var}(a_k) \hat{s_{ki}}^2} \tag{28}$$

Inspection of these voxelwise variance explained maps is very useful in searching through the estimated sources for application based relevance.

5.1 Correcting for Autocorrelation

Extending to the case of autocorrelated noise is straightforward. When $V_i \neq I$ then we proceed in an iterative fashion as follows:

1. Set $V_i = I$ and estimate $\hat{\bar{A}}$, $\hat{s}$ and compute the pointwise residual

$$\bar{r}_i = \bar{x}_i - \hat{\bar{\mu}} - \hat{\bar{A}}\hat{s}_i \tag{29}$$

2. Compute autocorrelation in $\bar{r}_i$ and prewhiten the data $\bar{x}_i$ using the estimated correlation matrix to get $\bar{x}_i^{pw}$. Run the PP algorithm on $\bar{x}_i^{pw}$ until $\bar{x}_i^{pw}$ does not change from one iteration to the next in an average sense. Various prewhitening schemes such as $AR(p)$ models or non-parametric approaches can be used. ADIS uses the non-parametric approach proposed in [38] to do prewhitening. By default ADIS will not do iterative prewhitening unless explicitly specified by the user.

6 Latent dimensionality estimation

Sophisticated bayesian strategies exist for estimating the latent dimensionality in the case of Gaussian sources [29]. In this section we develop a latent dimensionality estimation procedure that works very well both with Gaussian *and* non-Gaussian sources using a bootstrap/cross-validation procedure.

The latent dimensionality q is estimated in two steps. We first estimate a lower bound on the latent dimensionality (6.1) followed by a cross validation analysis to refine the lower bound (6.2). In subsection 6.3 we validate this approach via extensive numerical simulations.

6.1 Stage 1 - Estimating the lower bound

Let

$$X = [\bar{x}_1 - \hat{\bar{\mu}}, \bar{x}_2 - \hat{\bar{\mu}}, \ldots, \bar{x}_n - \hat{\bar{\mu}}] \tag{30}$$

and suppose $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p$ are the p eigenvalues of XX^T/n.

1. Randomly permute each column of the $p \times n$ matrix X to get the matrix X^b.
2. Compute the p eigenvalues λ_i^b of $X^b {X^b}^T/n$ such that $\lambda_1^b \geq \lambda_2^b \geq \ldots \lambda_p^b$
3. Choose q_l to be the largest value of i such that $\lambda_i > \lambda_i^b$.

First we destroy the systematic correlations between the columns of X via random permutations of each column. Then we estimate the singular values of the permuted covariance matrix and compare these with the singular values of the unpermuted covariance matrix. Only those singular values that exceed the ones from random permutation are deemed significant and the lower bound on latent dimensionality q_l is estimated to be the cardinality of those singular values.

If $P_i \in \mathbf{R}^{p \times p}$ are permutation matrices then we can write:

$$X^b = \frac{1}{n} \sum_{i=1}^{n} P_i (\bar{x}_i - \hat{\bar{\mu}})(\bar{x}_i - \hat{\bar{\mu}})^T P_i^T \tag{31}$$

Suppose

$$\bar{A} = U_a \Sigma_a V_a^T \tag{32}$$

is the singular value decomposition of A with $\Sigma_a = \mathrm{diag}(\sigma_{a_i})$

Let q be the true latent dimensionality. Then for large n it can be shown (and verified by simulation) that:

$$\lambda_i = \begin{cases} \sigma_{a_i}^2 + \sigma^2 & \text{if} \quad i \leq q \\ \sigma^2 & \text{if} \quad q+1 \leq i \leq (p-1) \\ 0 & \text{if} \quad i = p \end{cases} \tag{33}$$

and

$$\lambda_i^b = \begin{cases} \sigma^2 + \frac{1}{p-1}\sum_{i=1}^{q}\sigma_{a_i}^2 & \text{if} \quad i \leq (p-1) \\ 0 & \text{if} \quad i = p \end{cases} \tag{34}$$

Thus the non-zero eigenvalues of X^b satisfy $\lambda_i^b > \sigma^2$, the noise variance. Hence the largest index i such that $\lambda_i > \lambda_i^b$ is a lower bound for the latent dimensionality q.

6.2 Stage 2 - Cross Validation

Suppose the true latent dimensionality is assumed to be q. Then for large n the eigenvalues λ_i will follow equation (33) i.e, the eigenvalues $\{\lambda_i, i = (q+1)\ldots(p-1)\}$ should be well approximated by a constant σ^2. We estimate the quality of this model using leave one out cross validation [20]. Let M_q^{-k} be the mean of the eigenvalues λ_i from $q+1$ to $p-1$ excluding the index k.

$$M_q^{-k} = \frac{1}{p-2-q}\sum_{j=q+1, j\neq k}^{p-1}\lambda_i \tag{35}$$

The leave one out cross validation error assuming the true latent dimensionality to be q at point k is given by:

$$E(q,k) = \left(\lambda_k - M_q^{-k}\right)^2, k = q+1, \ldots, p-1 \tag{36}$$

The mean cross validation error and its variance can be estimated from these pointwise values as follows:

$$\bar{E}(q) = \frac{1}{p-1-q}\sum_{k=q+1}^{p-1}E(q,k) \tag{37}$$

$$\mathrm{Var}(\bar{E}(q)) = \frac{1}{p-1-q}\mathrm{Var}\{E(q,k), k = q+1, \ldots, p-1\} \tag{38}$$

When q is smaller than q_{true} then both $\bar{E}(q)$ and $\mathrm{Var}(\bar{E}(q))$ will be large and when q is greater than q_{true} then both $\bar{E}(q)$ and $\mathrm{Var}(\bar{E}(q))$ will be small. Since the eigenvalues are expected to remain constant beyond q_{true} the change in $\bar{E}(q)$ will be small beyond q_{true}. We define the following index for a given value of q quantifying the change in cross validation error from q to $q+1$.

$$\Delta(q) = \frac{\bar{E}(q) - \bar{E}(q+1)}{\sqrt{\mathrm{Var}(\bar{E}(q)) + \mathrm{Var}(\bar{E}(q+1))}} \tag{39}$$

If q_{true} is the true latent dimensionality then $\Delta(q)$ will tend to have a maximum at $q_{true}-1$ since this is the point which will show the largest change in cross validation error in going from $q_{true}-1$ to q_{true}. We thus propose the estimate

$$q_{true} = 1 + \underset{q}{\operatorname{argmax}}\, \Delta(q), q = q_l, \ldots, p-4 \tag{40}$$

where q_l is a lower bound on q calculated from stage 1.

The estimation of $\bar{E}(q)$ uses a smaller number of points when q gets very close to $p-1$. Thus the estimate $\Delta(q)$ becomes unstable when q is within a few time points of $p-1$. In order to robustify our estimate against this instability we define the cumulative maximum index function which calculates the index of maximum of $\Delta(q)$ from $q = q_l, \ldots, r$.

$$f(r) = \underset{q}{\operatorname{argmax}}\, \Delta(q), q = q_l, \ldots, r \tag{41}$$

Then we count the number of times that a maximum is detected at y

$$g(y) = \text{Card}\{r : f(r) = y\} \tag{42}$$

and define the estimate of dimensionality to be

$$q_{true} = 1 + \underset{y}{\operatorname{argmax}}\, g(y) \tag{43}$$

6.3 Validation of the approach

To test this latent dimensionality algorithm we generate data as per equation (1). Details of the simulation are as follows:

1. The true mixing matrix A was chosen to be a $p \times q$ matrix where the elements were drawn from a uniform distribution in (0,1). This random matrix was then scaled so that its minimum singular value $\sigma_{min}(A) = 1$.
2. The sources s in (1) were generated from three different types of distributions based on their kurtosis "excess" values γ . The chosen distributions were Gaussian ($\gamma = 0$), Uniform ($\gamma < 0$) and Gamma ($\gamma > 0$).
3. The ratio of noise standard deviation in (1), σ to the minimum singular value of A, $\frac{\sigma_{min}(A)}{\sigma}$ was varied between $0.75, 1, \ldots, 2$.
4. The ratio of the true latent dimensionality to the dimensionality of the observed data $\frac{q_{true}}{p}$ was varied between 0.1,0.2,. . . ,0.5.

5. This process was repeated 20 times for each combination of source type, $\frac{\sigma_{min}(A)}{\sigma}$ ratio and $\frac{q_{true}}{p}$ ratio.
6. For each individual simulation we estimated the latent dimensionality $\hat{q}$ based on the 2 stage estimation strategy described above.

We chose $p = 50$ and $n = 1000$ as fixed parameters of the simulation. Simulations show that this 2 stage estimation is almost unbiased for both Gaussian and non-Gaussian embedded sources for all values of the ratios $\frac{\sigma_{min}(A)}{\sigma}$ and $\frac{q_{true}}{p}$. Results are shown in figure 1(a) - 1(c).

7 Benchmarking: Comparison with other BSS algorithms

Choosing the negative entropy index as our objective function and without imposing any additional constraints, our algorithm attempts to estimate sources that are independent. To test and compare our algorithm with others, we used an approximation to negative entropy as proposed in [26] (see appendix for details).

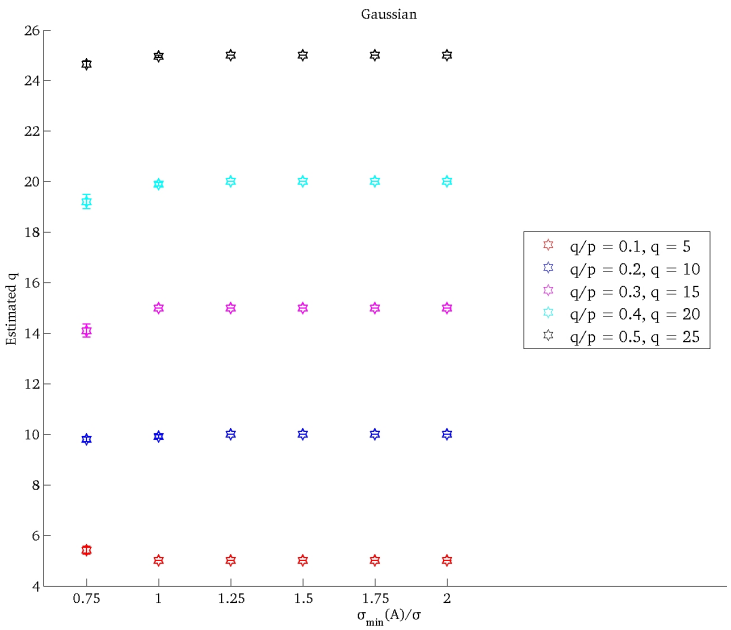
ICALAB [8], [7] (available from `http://www.bsp.brain.riken.go.jp/ICALAB/`) is a Matlab package for comparing algorithms for BSS. We used ICALAB to compare the performance of our algorithm with other standard BSS algorithms such as FJADE [6], FPICA [26], [24], EFICA [27], [28] , ERICA [13] and UNICA [14] which use higher order statistics to separate sources.

The quality of source extraction is measured using the Source to Interferences Ratio (SIR) [37] (of the estimated mixing matrix) which measures the ratio of the energy of the estimated source projected onto the true source to the energy of the estimated source projected onto the other sources. Higher values of (SIR) indicate better performance. Please see the appendix for details. ICALAB also comes with standard benchmarking datasets (`http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/`).

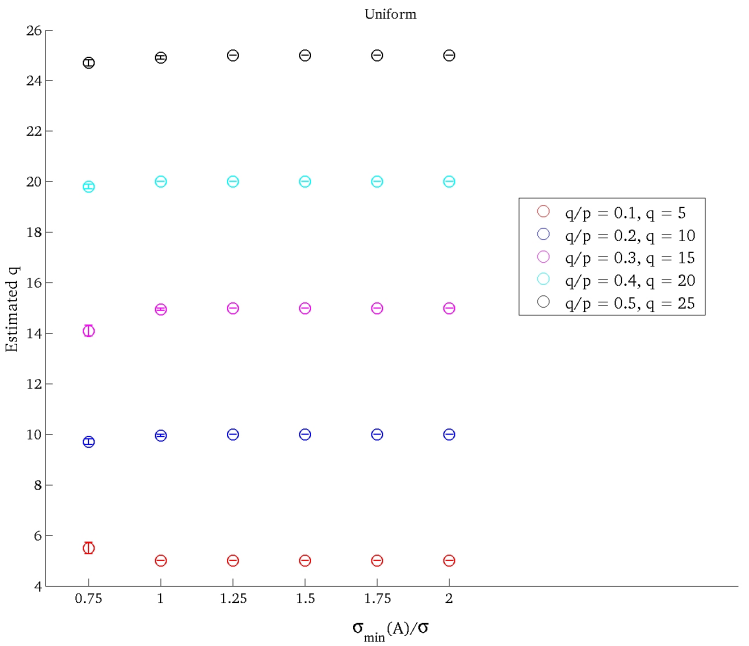
A Monte Carlo analysis was performed using ICALAB by generating uniformly distributed random matrices A_i and creating a mixed source data-set X_i for a given set of sources S.

$$X_i = A_i\, S,\ i = 1, 2, \ldots, n_b \tag{44}$$

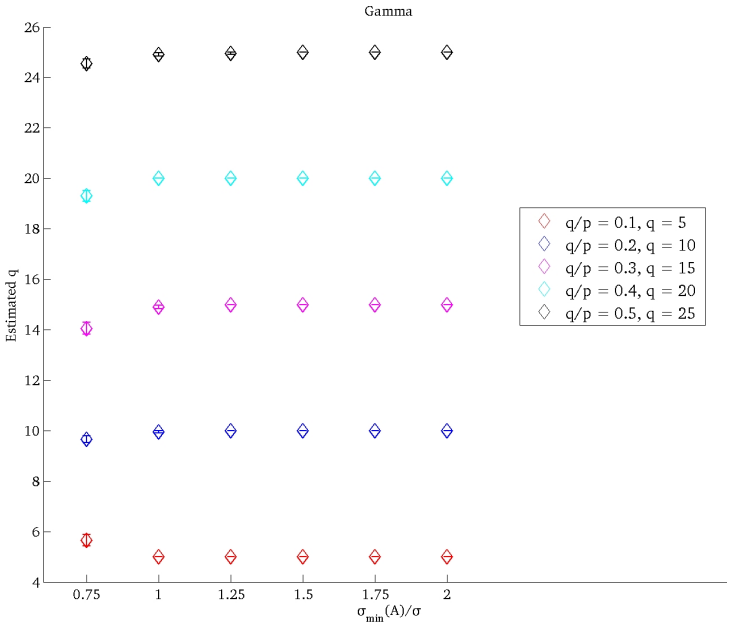
To get a baseline measure of performance for each algorithm, we use square mixing without additional noise for the simulation. Each algorithm was then run on this mixed data set. This process was repeated $n_b = 100$ times for each dataset using a ***new*** mixing matrix every time. In ICALAB, most of the algorithms are given default parameters that are tuned optimum values for typical data. As suggested in ICALAB, we use these default algorithmic parameters for benchmarking purposes.



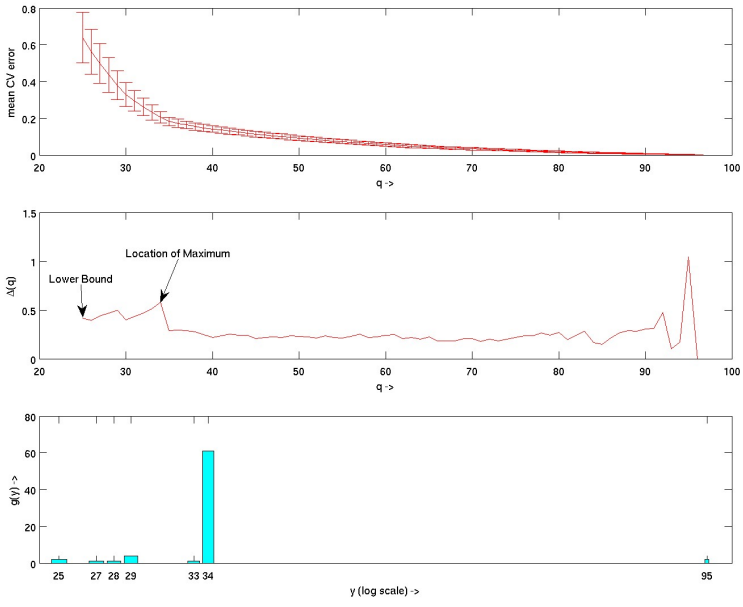
(a) Gaussian sources (kurtosis excess = 0)



(b) Uniform sources (kurtosis excess < 0)



(c) Gamma sources (kurtosis excess > 0)



(d) Illustration of latent dimensionality estimation

Figure 1: (a), (b) and (c) depict simulations showing the performance of latent dimensionality estimation procedure on various source types at different $\frac{\sigma_{min}(A)}{\sigma}$ ratios parameterized by various q/p ratios. (d) Latent dimensionality estimation for a Gaussian sources with $q_{true} = 35$, $p = 100$, $n = 1000$ and $\frac{\sigma_{min}(A)}{\sigma} = 0.75$. $\Delta(q)$ attains a maximum for $q = 34$ and so $\hat{q} = 1 + 34 = 35$

ADIS is able to perform non-square BSS in the presence of noise. However, since the other algorithms in our benchmarking test have not been designed to do this, we think its unfair to compare non-square ability of ADIS with other algorithms.

The 13 benchmarking datasets and their short descriptions are given in the appendix. In order to evaluate the effect of different types of mixing matrices, we ran additional Monte Carlo simulations when A was chosen to be one of the following (a) Random sparse (b) Random bipolar (c) Symmetric random (d) Ill conditioned random (e) Hilbert (f) Toeplitz (g) Hankel (h) Orthogonal (i) Nonnegative symmetric (j) Bipolar symmetric (k) Skew symmetric.

The results are shown in figures 2(a) - 4(d) and table 1. The key performance feature is the SIR index [37] (see appendix for definition), the higher the value the better. Another important feature is the variability of SIR over 100 mixtures each generated using a different mixing matrix but containing the same underlying sources. Ideally an algorithm should be robust enough to converge to the same solution regardless of variation in the mixing matrix. The standard deviation of SIR captures this variability, the lower the variability of SIR the better. Additional results showing simulation results with different types of mixing matrices A are shown in 7(a) - 9(d).

ADIS perfomed better than all other algorithms in almost all cases both in terms of the mean SIR index as well as the standard deviation of the mean SIR, which measures the robustness and convergence to the same solution. ADIS also produces extensive convergence diagnostics a sample of which is shown in figure 6(a). These diagnostics guarantee convergence and improve confidence in the estimated sources.

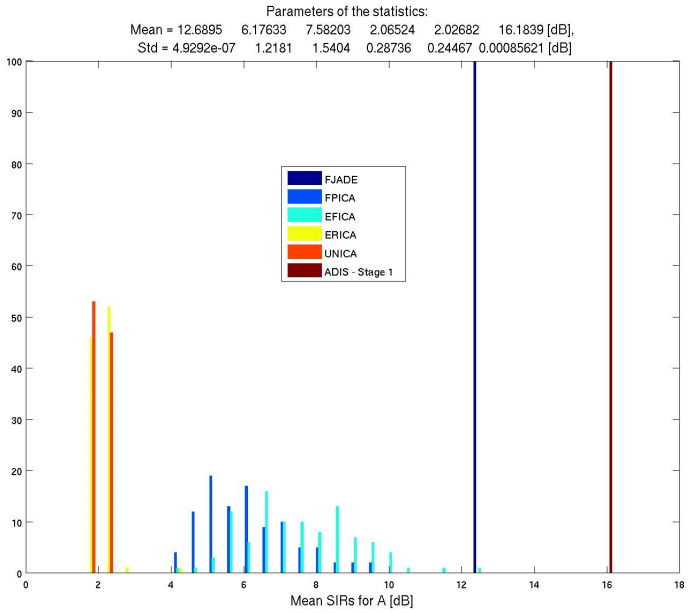
An illustration of performance improvement using ADIS (Stage 1 + 2) over Stage 1 is shown fir the ACsparse10 dataset in figure 5(b). The corresponding convergence diagnostics are shown in figure 6(b). ADIS Stage 2 performs better than ADIS Stage 1 but we did not observe as dramatic an improvement as seen for ACsparse10. Other ICA algorithms were outperformed using only ADIS Stage 1.

All experiments were performed on a computer with an Intel Xeon (TM) processor (3.4 GHz) and 4GB of RAM. The runtimes of ADIS (Stage 1) were comparable to those of FPICA, ERICA and UNICA. We found EFICA and JADE to be faster than other algorithms in general.

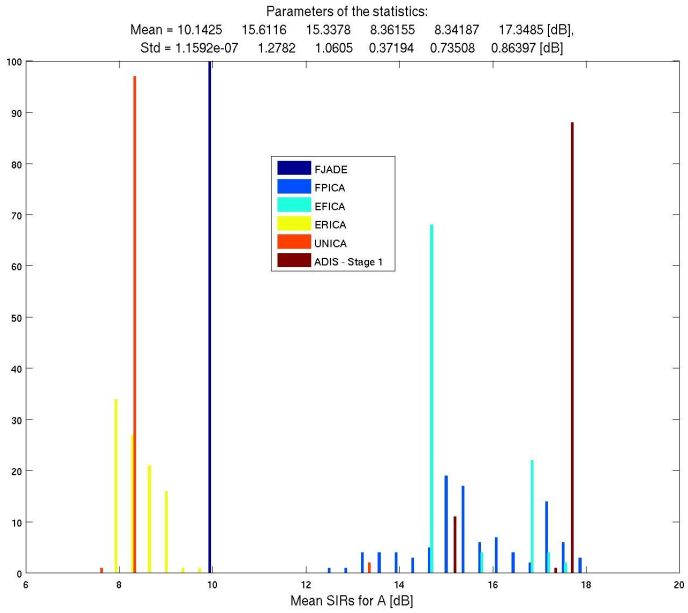
8 Case Study using real fMRI data

8.1 fMRI case study

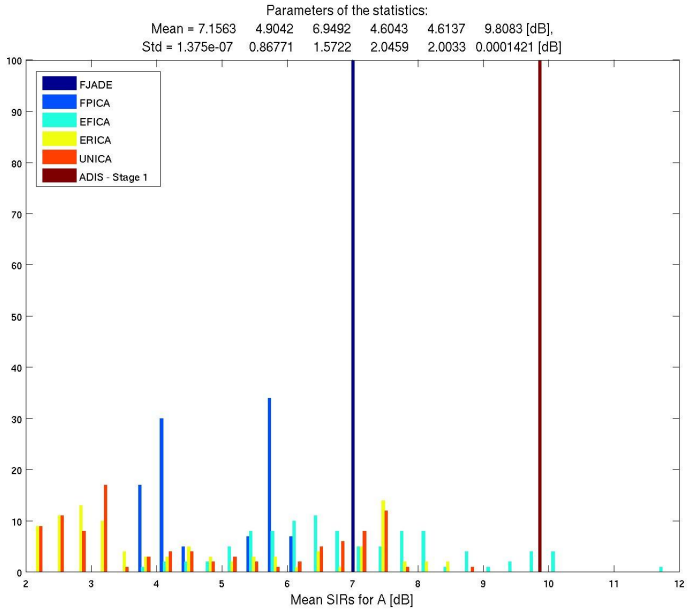
To demonstrate how ADIS performs on a real dataset, we used the "FSL Evaluation and Example Data Suite" (FEEDS) from FMRIB Image Analysis Group, Oxford University. The URL for this data suite is: `http://www.fmrib.ox.ac.uk/fsl/feeds/doc/index.html` One of the datasets in the example suite contains an audio visual experiment with two explanatory variables, the visual



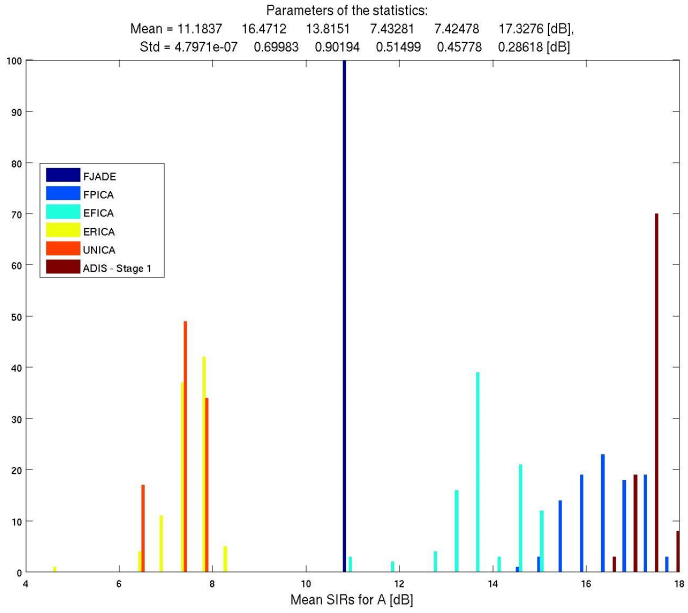
(a) nband5



(b) 10halo

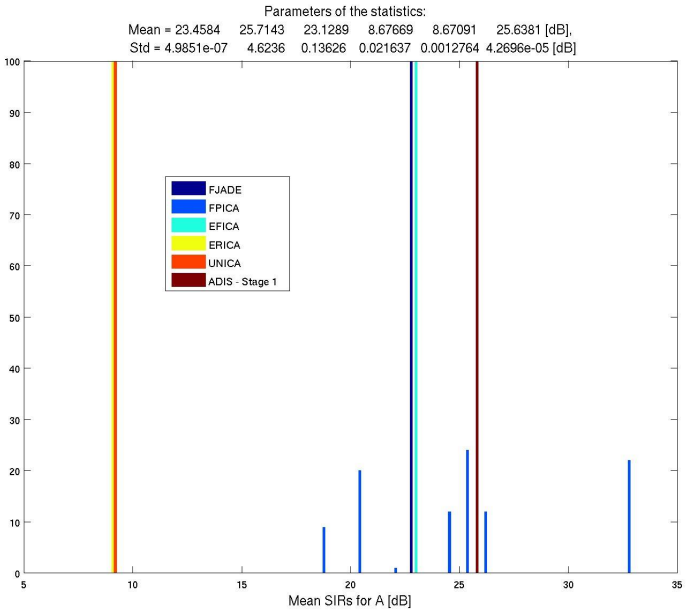


(c) GnBand

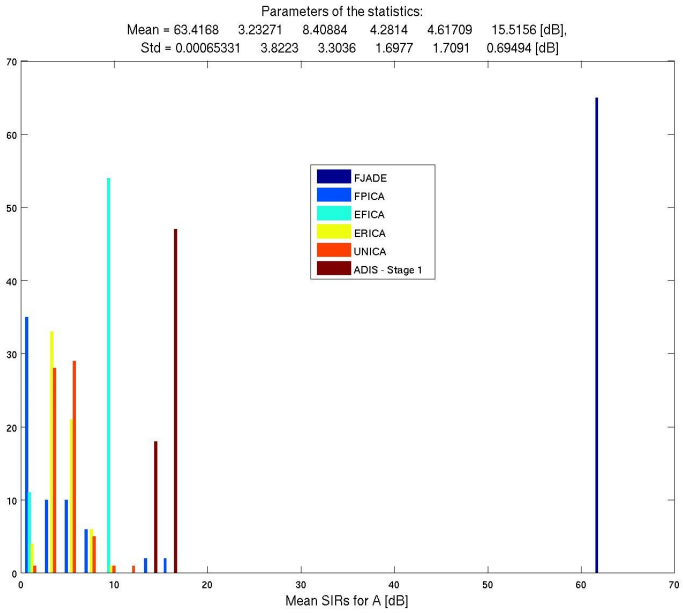


(d) acspeech16

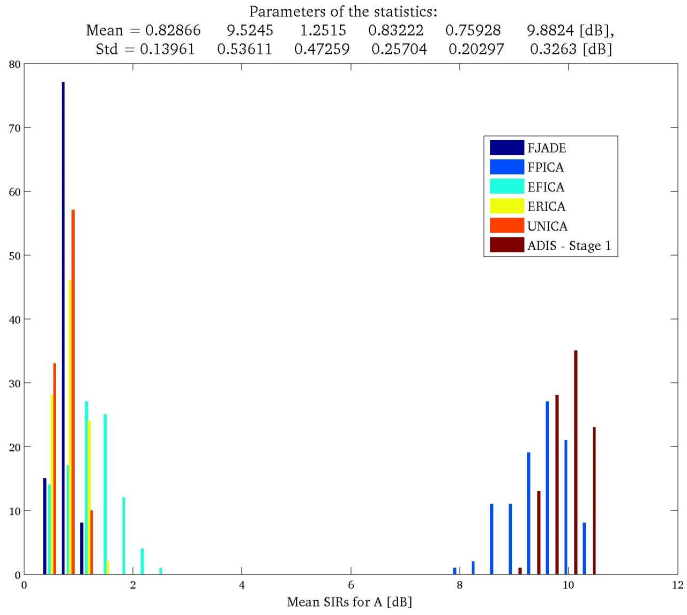
Figure 2: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices for various benchmarking datasets.



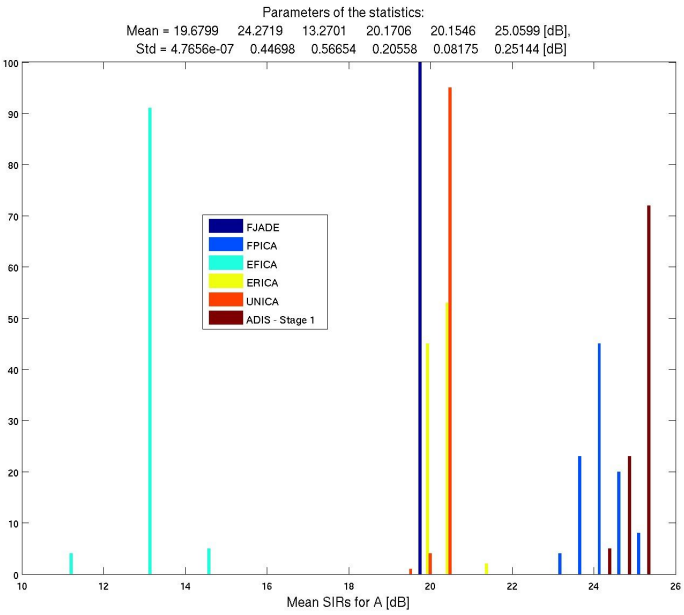
(a) ABio5



(b) ACsparse10

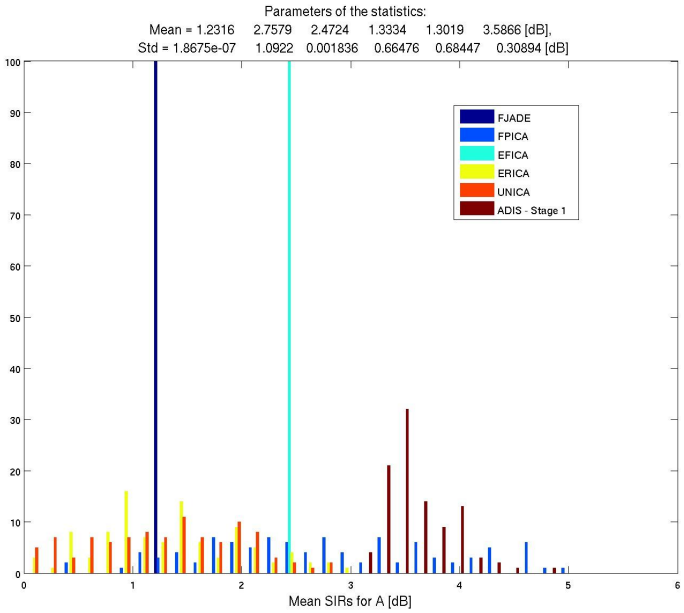


(c) 25SpeakersHALO

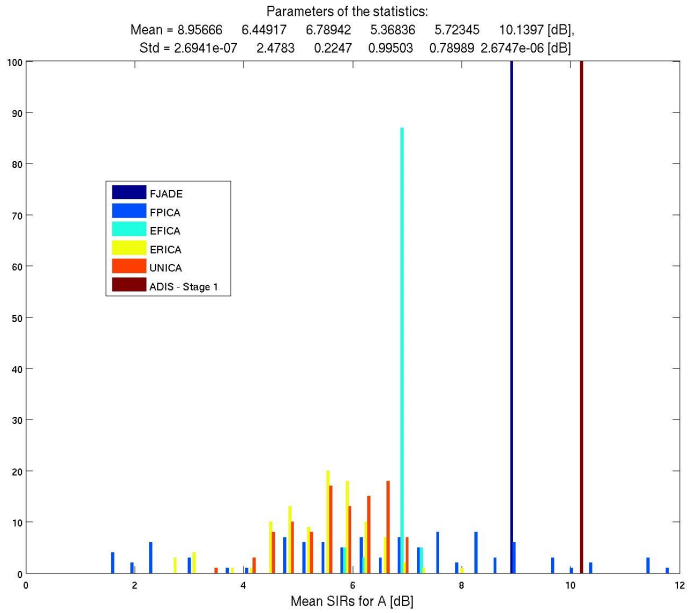


(d) Vsparserand10

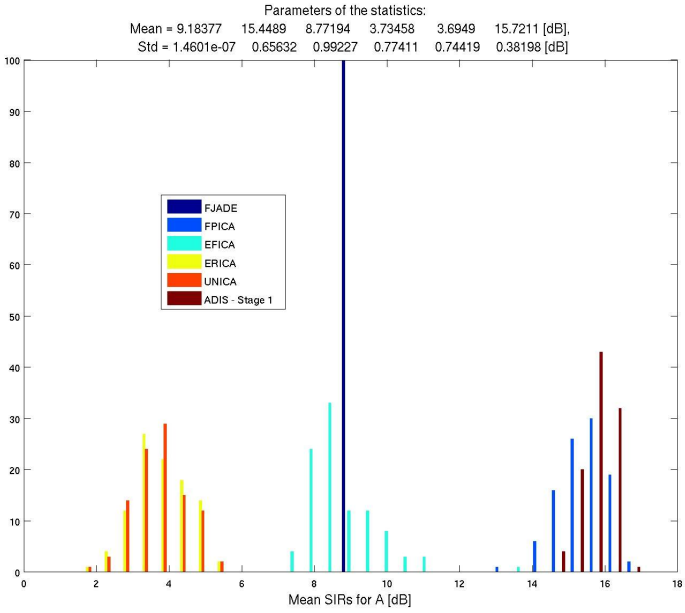
Figure 3: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices for various benchmarking datasets.



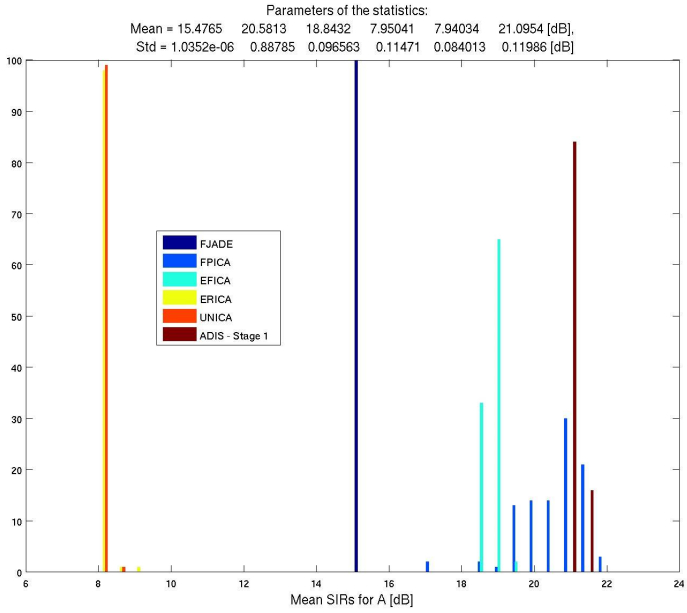
(a) ACsincpos10



(b) X5smooth



(c) speech20

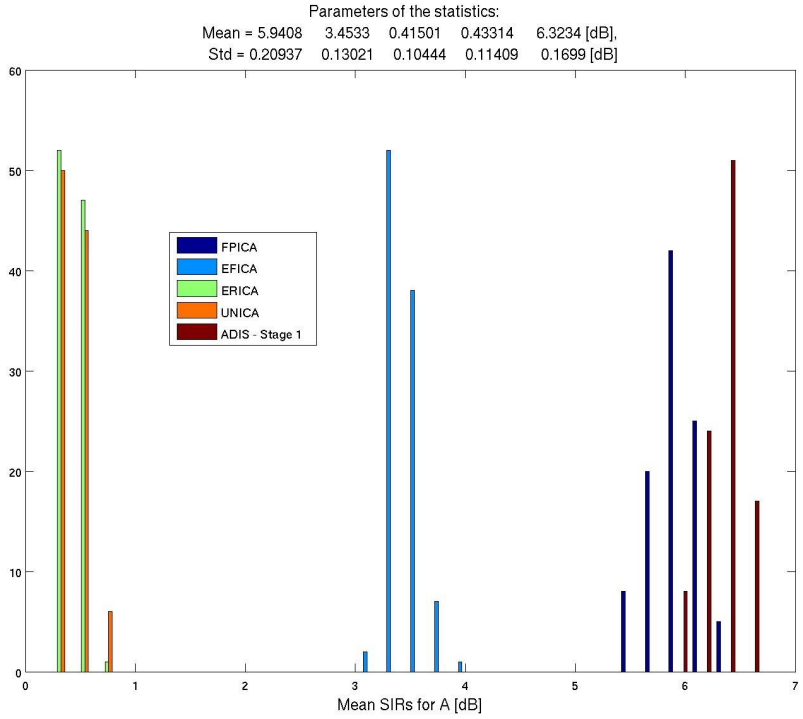


(d) X10randsparse

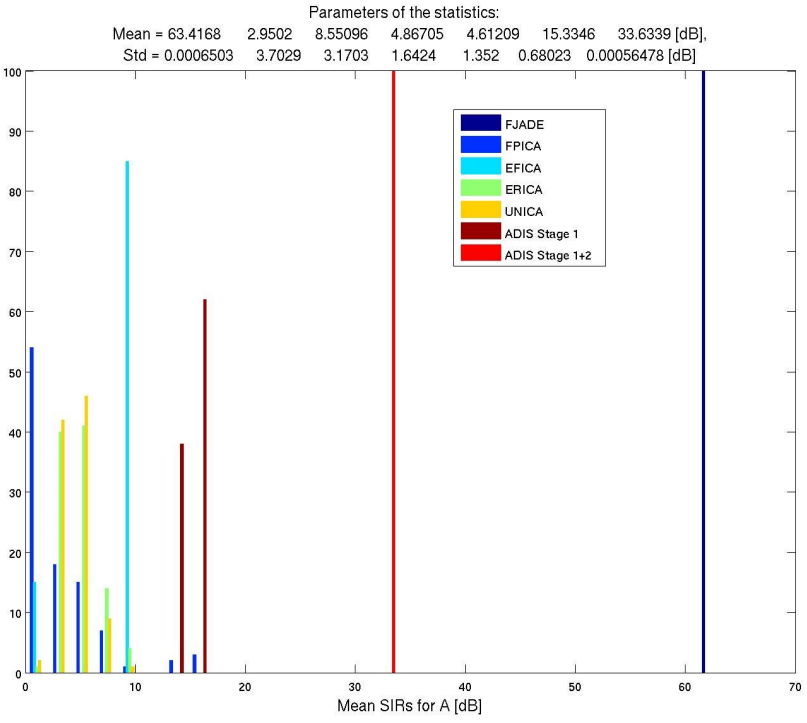
Figure 4: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices for various benchmarking datasets.

		Algorithm, $M = S\bar{I}R$ [dB] and $S = std(S\bar{I}R)$ [dB]					
Dataset		FJADE	FPICA	EFICA	ERICA	UNICA	ADIS - Stage 1
nband5	M	12.6895	6.17633	7.58203	2.06524	2.02682	16.1839
	S	4.9292e−07	1.2181	1.5404	0.28736	0.24467	8.5621e−04
10halo	M	10.1425	15.6116	15.3378	8.36155	8.34187	17.3485
	S	1.1592e−07	1.2782	1.0605	0.37194	0.73508	0.86397
GnBand	M	7.1563	4.9042	6.9492	4.6043	4.6137	9.8083
	S	1.375e−07	0.86771	1.5722	2.0459	2.0033	1.421e−04
acspeech16	M	11.1837	16.4712	13.8151	7.43281	7.42478	17.3276
	S	4.7971e−07	0.69983	0.90194	0.51499	0.45778	0.28618
ABio5	M	23.4584	25.7143	23.1289	8.67669	8.67091	25.6381
	S	4.9851e−07	4.6236	0.13626	0.021637	1.2764e−03	4.2696e−05
ACsparse10	M	63.4168	3.23271	8.40884	4.2814	4.61709	15.5156
	S	6.5331e−04	3.8223	3.3036	1.6977	1.7091	0.69494
25SpeakersHALO	M	0.82866	9.5245	1.2515	0.83222	0.75928	9.8824
	S	0.13961	0.53611	0.47259	0.25704	0.20297	0.3263
VSparserand10	M	19.6799	24.2719	13.2701	20.1706	20.1546	25.0599
	S	4.7656e−07	0.44698	0.56654	0.20558	0.08175	0.25144
sincpos10	M	1.2316	2.7579	2.4724	1.3334	1.3019	3.5866
	S	1.8675e−07	1.0922	0.01836	0.66476	0.68447	0.30894
X5smooth	M	8.95666	6.44917	6.78942	5.36836	5.72345	10.1397
	S	2.6941e−07	2.4783	0.2247	0.99503	0.78989	2.6747e−06
Speech20	M	9.18377	15.4489	8.77194	3.73458	3.6949	15.7211
	S	1.4601e−07	0.65632	0.99227	0.77411	0.74419	0.38198
X10randsparse	M	15.4765	20.5813	18.8432	7.95041	7.94034	21.0954
	S	1.0352e−06	0.88785	0.096563	0.11471	0.084013	0.11986
64soundstd	M	x	5.9408	3.4533	0.41501	0.43314	6.3234
	S	x	0.20937	0.13021	0.10444	0.11409	0.1699

Table 1: Mean SIR (M) and its standard deviation (S) for various benchmark datasets over 100 Monte Carlo simulations for different algorithms. The benchmark datasets are a part of ICALAB [8], [7]. An 'x' means that the algorithm failed to converge. The algorithms were ranked based on not only their mean SIR (M) and standard deviation (S) but also the entire SIR histogram. The color red is used to mark the best performing algorithm and the color blue is used to mark the 2nd best. In cases where more than one algorithm is marked with the same color, both algorithms were judged to perform equally well.

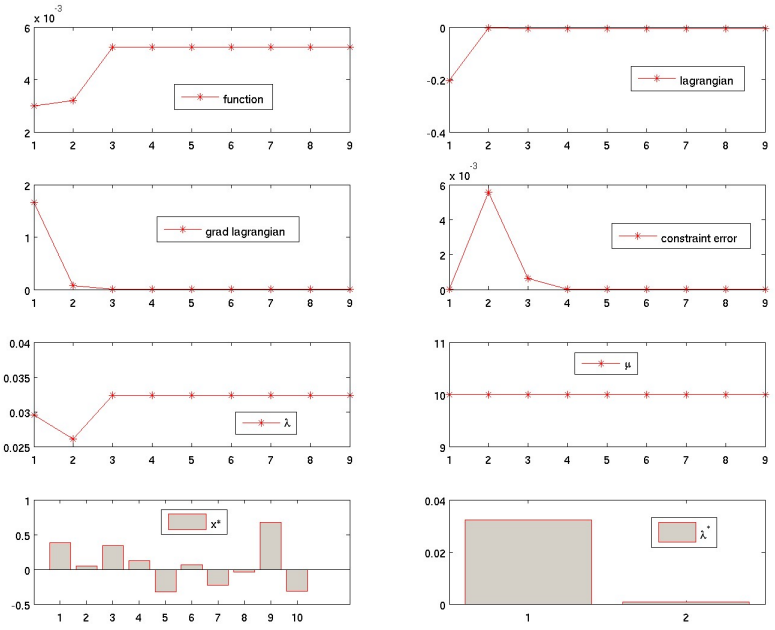


(a) 64sounds mean SIR

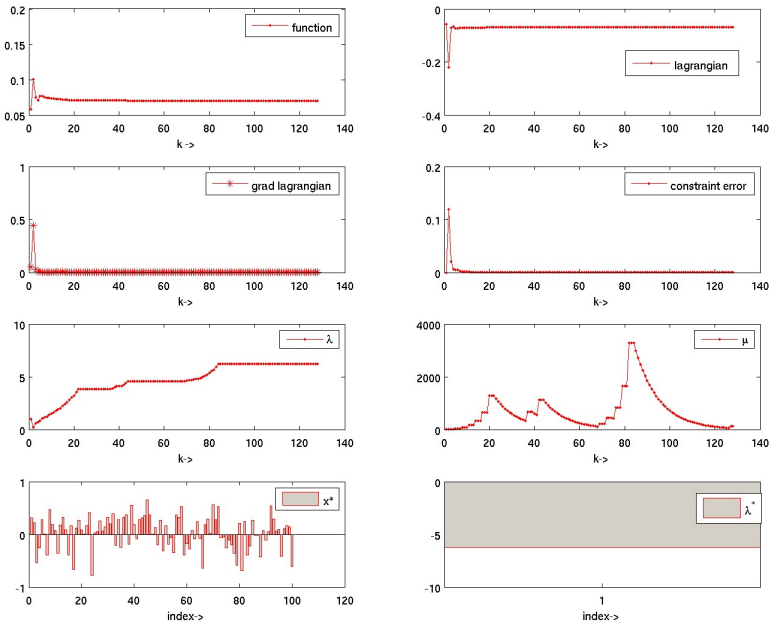


(b) ACsparse10 ADIS Stage 1+2

Figure 5: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices on (a) the 64sounds benchmark dataset. This is a relatively large dataset with 64 sources. FJADE fails on this test case. (b) the ACsparse10 benchmarking dataset. This figure illustrates the improvement in SIR using the joint optimization of ADIS Stage 1+2 over ADIS Stage 1

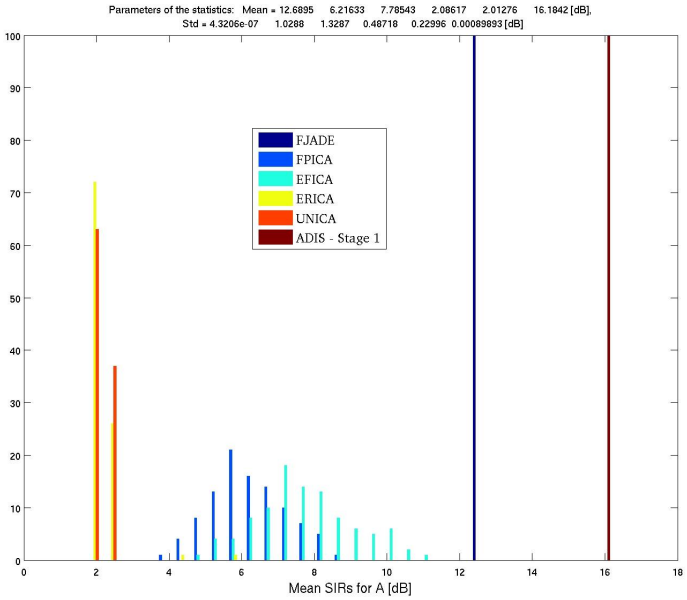


(a) convergence diagnostics

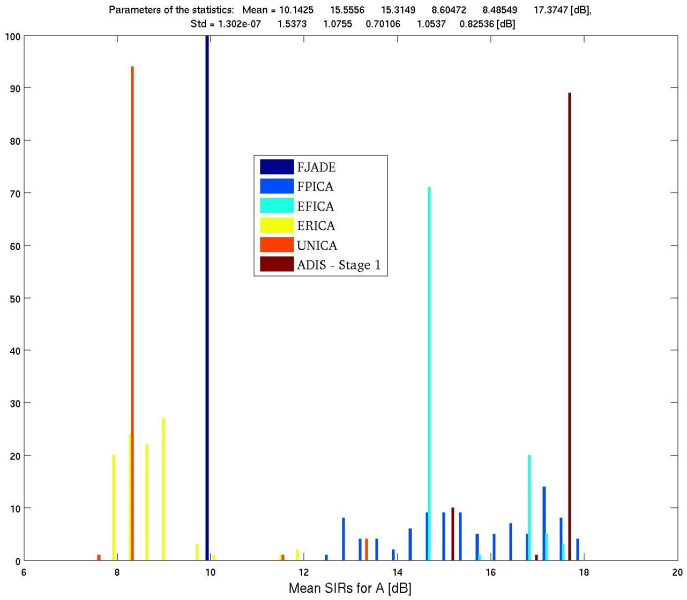


(b) 2 stage convergence diagnostics

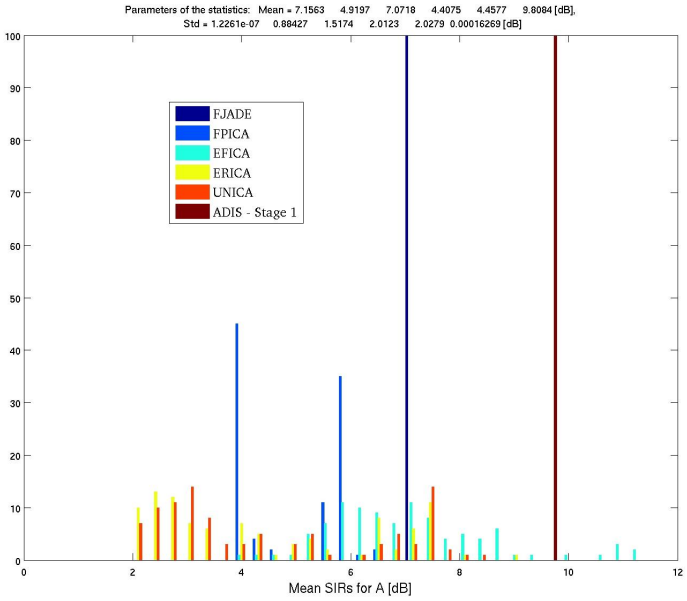
Figure 6: (a) Sample Convergence Diagnostic plot for the 10halo benchmark in estimating the 2nd component. Figure shows the evolution of objective function, the Lagrangian, norm of the gradient of the Lagrangian, norm of the constraint satisfaction error, norm of the Lagrange multipliers and penalty parameter over algorithm iterations. ADIS *guarantees* the optimality of the estimated sources. (b) Convergence diagnostics for ADIS Stage 1+2 for ACsparse10 dataset. The joint optimization was initialized using the optimal solution from ADIS Stage 1.



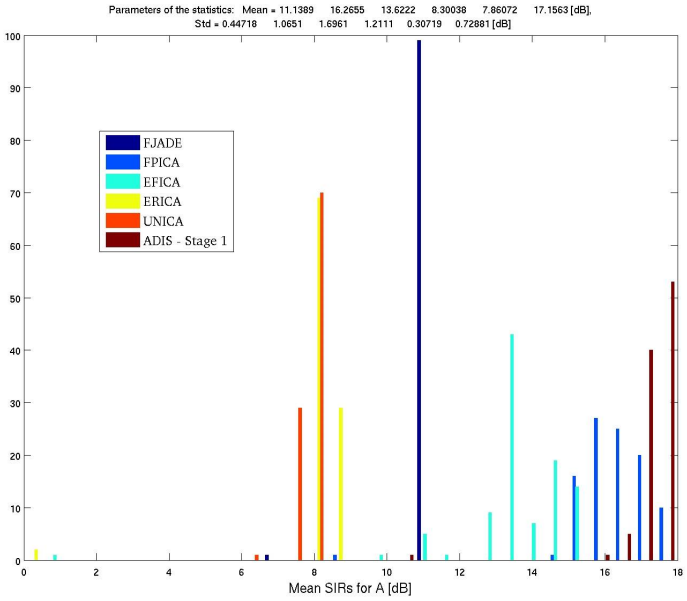
(a) nband5



(b) 10halo

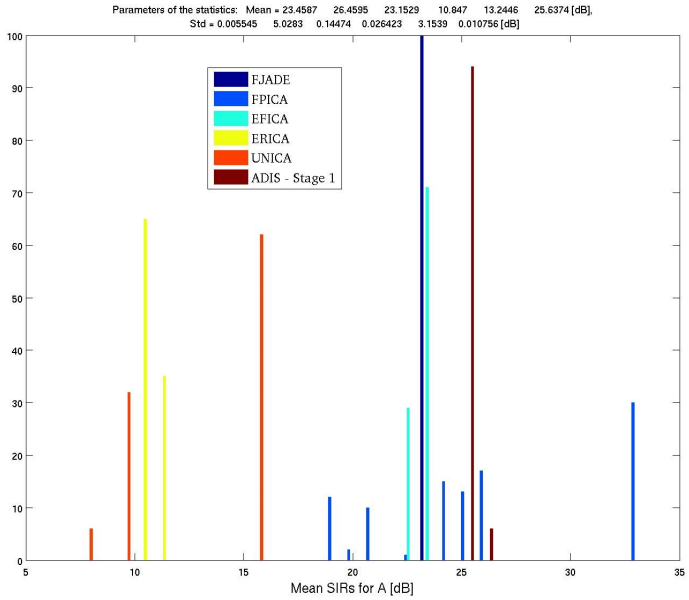


(c) GnBand

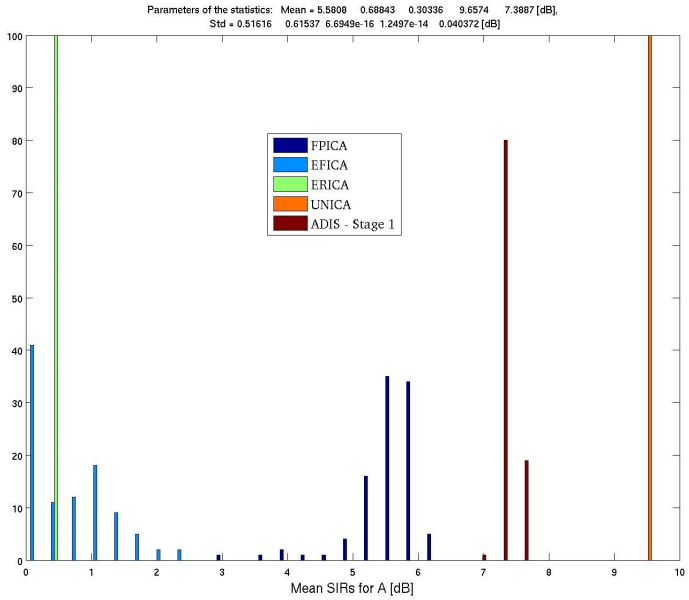


(d) acspeech16

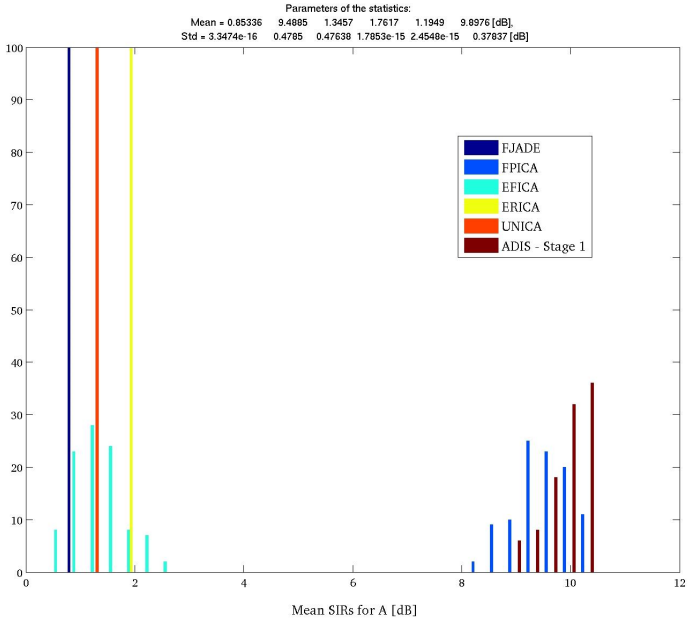
Figure 7: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices for various benchmarking datasets. The properties of mixing matrices A used for the simulations are as follows: (a) Random sparse (b) Random sparse (c) Random bipolar (d) Symmetric random



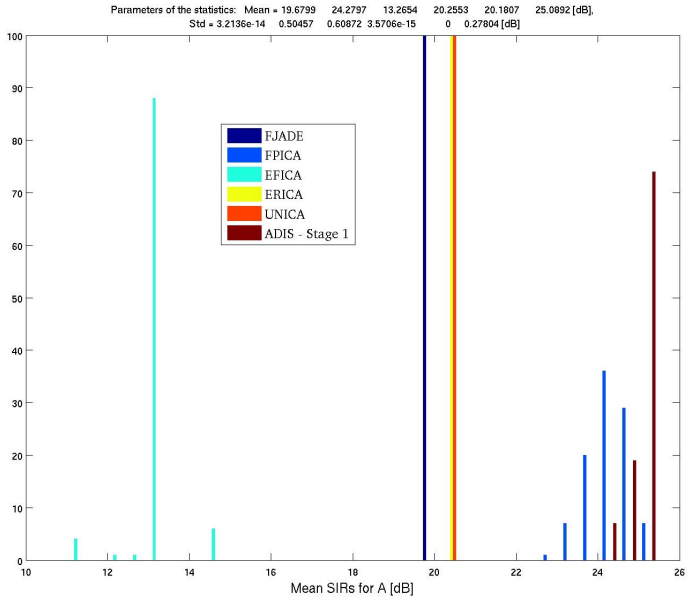
(a) ABio5



(b) ACsparse10

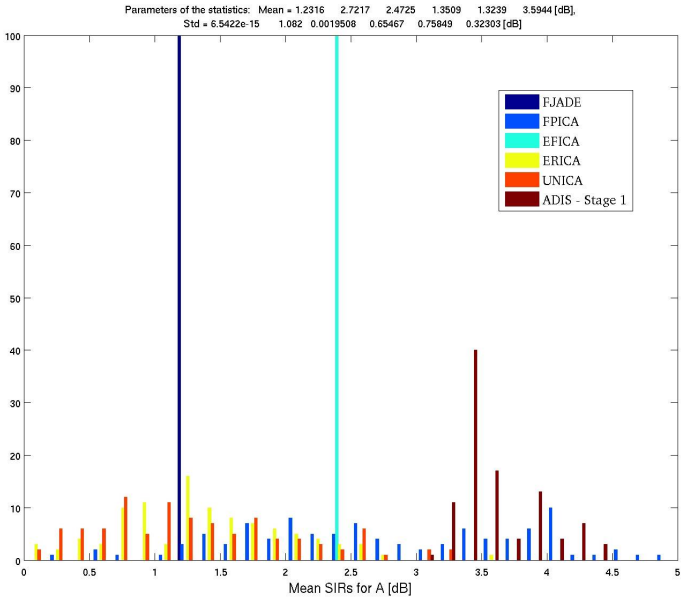


(c) 25SpeakersHALO

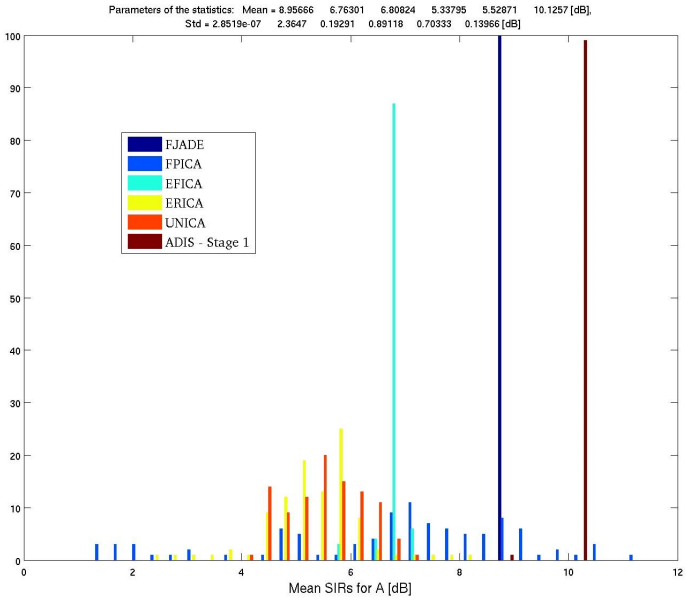


(d) Vsparserand10

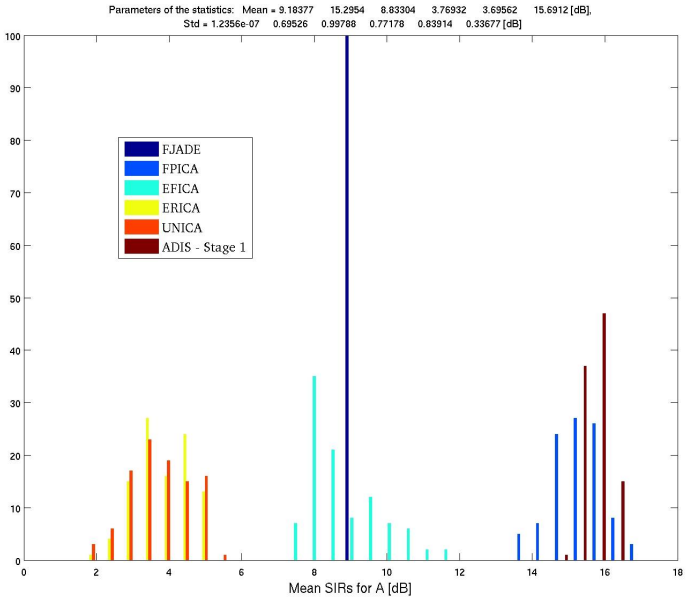
Figure 8: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices for various benchmarking datasets. The properties of mixing matrices A used for the simulations are as follows: (a) Ill conditioned random (b) Hilbert (c) Toeplitz (d) Hankel



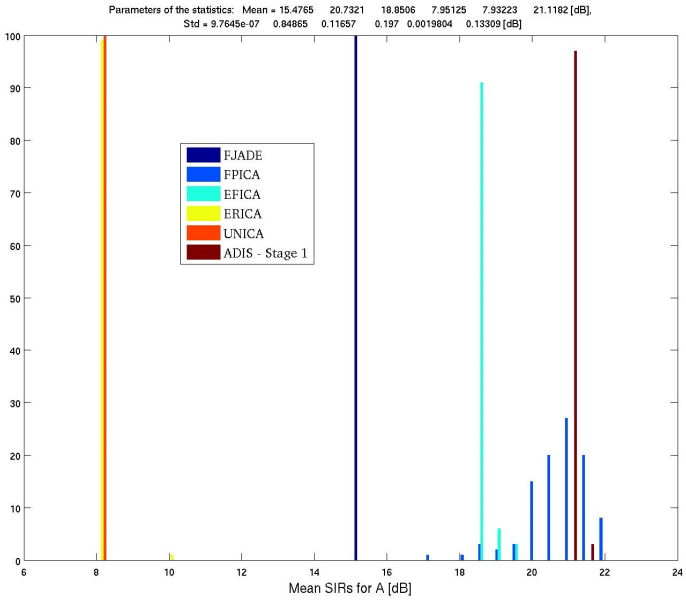
(a) ACsincpos10



(b) X5smooth



(c) speech20



(d) X10randsparse

Figure 9: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices for various benchmarking datasets. The properties of mixing matrices A used for the simulations are as follows: (a) Orthogonal (b) Nonnegative symmetric (c) Random bipolar (d) Skew symmetric

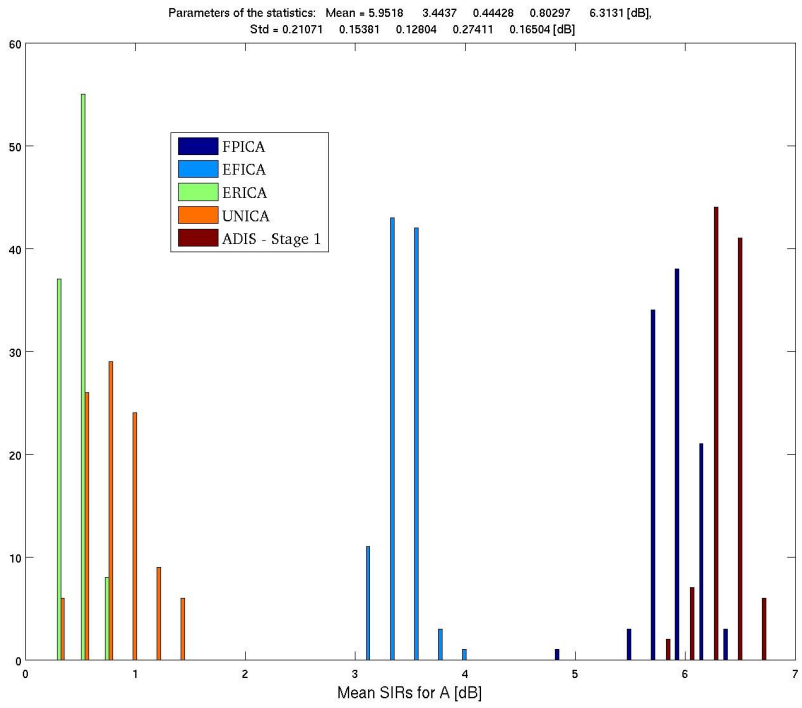


Figure 10: Histograms of mean SIR for each algorithm over 100 Monte Carlo simulations using randomly generated mixing matrices on the 64sounds benchmark dataset. This is a relatively large dataset with 64 sources. FJADE fails on this test case. A was chosen to be an ill conditioned random mixing matrix for this simulation.

stimulus (30s off, 30s on) and an auditory stimulus (45s off, 45s on). Analysis was carried out using FEAT (FMRI Expert Analysis Tool) Version 5.4, part of FSL (FMRIB's Software Library).

`www.fmrib.ox.ac.uk/fsl`

The following pre-statistics processing was applied; motion correction using MCFLIRT [Jenkinson 2002]; non-brain removal using BET [Smith 2002]; spatial smoothing using a Gaussian kernel of FWHM 5mm; mean-based intensity normalisation of all volumes by the same factor; highpass temporal filtering (Gaussian-weighted LSF straight line fitting, with sigma=50.0s).

ADIS estimated a latent dimensionality of $q = 34$. The source components activating the auditory and visual cortex were identified by inspecting the estimated voxelwise variance explained map. The results are shown in figures 11-12.

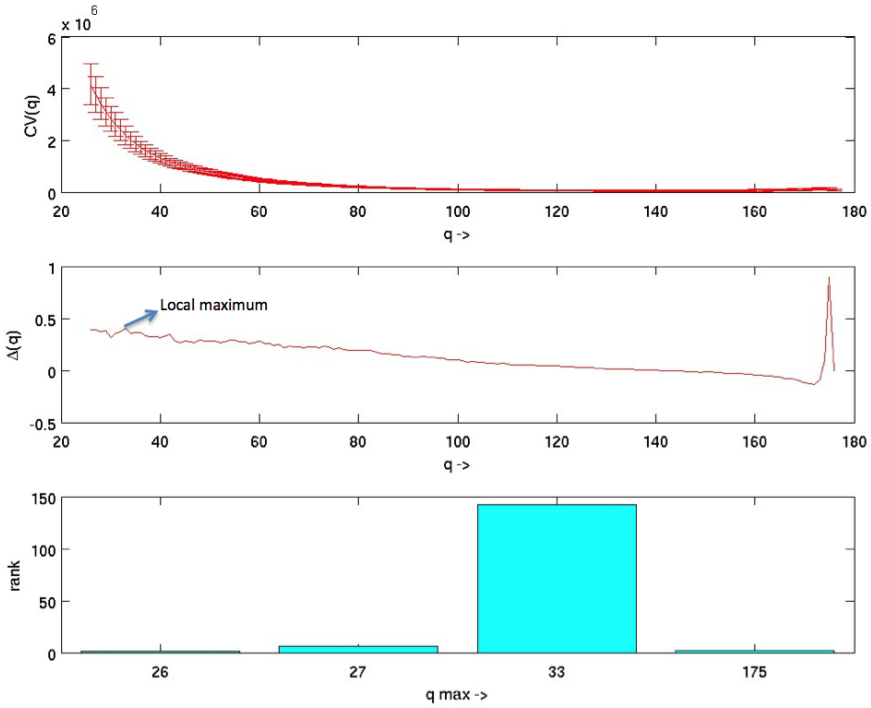
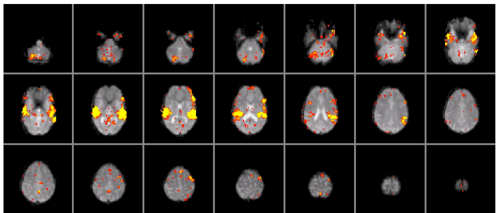
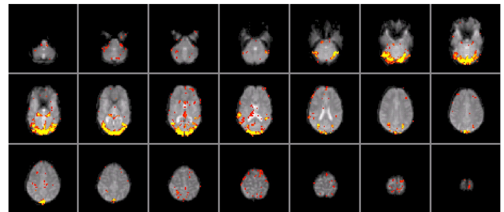


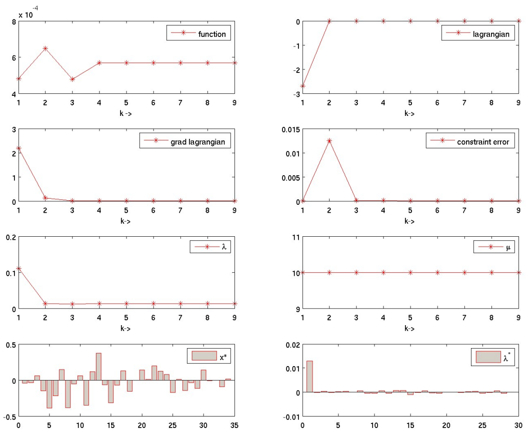
Figure 11: Latent dimensionality estimation summary for fMRI data. The number of latent sources were estimated to be $q = 34$.



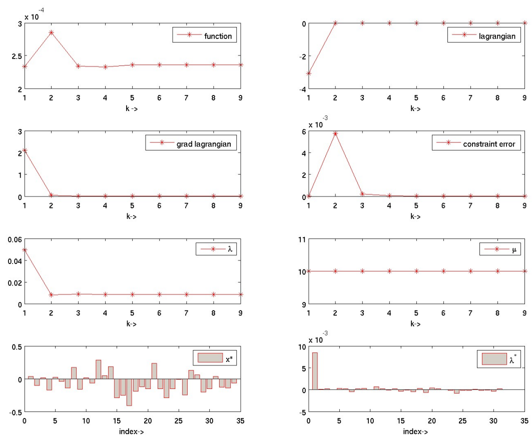
(a) Estimated Auditory Cortex activating source z-statistic for the audio-visual fMRI data thresholded at $z > 5$



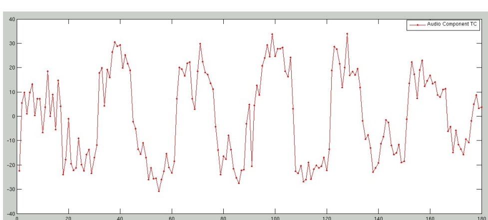
(b) Estimated Visual Cortex activating source z-statistic for the audio-visual fMRI data thresholded at $z > 5$



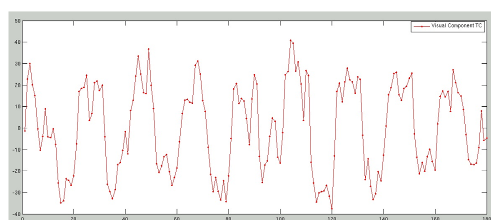
(c) Convergence Diagnostic plot for the Auditory source estimation



(d) Convergence Diagnostic plot for the Visual source estimation



(e) Associated Timecourse (from mixing matrix) for the Auditory source



(f) Associated Timecourse (from mixing matrix) for the Visual source

Figure 12: Result of applying ADIS (Stage 1) to audio-visual fMRI data

9 Conclusion

We implemented ADIS, an algorithm for probabilistic, constrained and non-square projection pursuit. We validated all aspects of ADIS including the latent dimensionality estimation procedure and its optimization core. When compared to other algorithms using standard benchmarking datasets using ICALAB, we find our algorithm outperforms other standard algorithms such as FastICA, FPICA, JADE, ERICA and UNICA in terms of both robustness and separation quality. Our algorithm also guarantees "optimality" for each blind source via extensive convergence diagnostics and enables the user to use arbitrary contrast function and constraints for BSS. We hope it will be useful as a general BSS tool for the signal processing and fMRI community.

10 Appendix

11 Negentropy Index

Given a random variable X, the negative entropy a measure of non-Gaussianity. It is easy to show that imposition of independence on sources in BSS is equivalent to maximization of negentropy.

Robust approximations to negative entropy were developed in [23]. If G is a non-quadratic, nonlinear function and v is a Gaussian random variable of the same variance as X then the negentropy measure $J(X)$ is given as [26]

$$J(X) \propto [E(G(X)) - E(G(v)]^2 \tag{45}$$

In this paper, we used the following function [23] for G:

$$G(x) = \log [\cosh (x)] \tag{46}$$

where $\cosh (x)$ is the hyperbolic cosine function

$$\cosh (x) = \frac{e^x + e^{-x}}{2} \tag{47}$$

12 Sources to Interferences Ratio (SIR)

The SIR ratio is defined in [37]. We give here a brief summary of the key equations from that paper. Let $y = \{y_1, y_2, \ldots, y_k\}$ and let P_y be the orthogonal projector onto the subspace spanned

by $y_1, y_2, \ldots y_k$. If $s = [s_1, s_2, \ldots, s_k]$ are the true sources and if $\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_k$ are the corresponding estimated values then define:

$$s_{target} = P_{s_j} \hat{s}_j \tag{48}$$

$$e_{interf} = P_s \hat{s}_j - P_{s_j} \hat{s}_j \tag{49}$$

The purity of source separation is measured using the SIR performance index defined as follows:

$$SIR = 10 \log_{10} \frac{||s_{target}||^2}{||e_{interf}||^2} \tag{50}$$

13 Benchmarking datasets

The 13 benchmarking datasets and their short descriptions are as follows:
(`http://www.bsp.brain.riken.jp/ICALAB/ICALABSignalProc/`) :

- nband5 - contains 5 narrow band sources. This is a rather "easy" benchmark for second order separation algorithms but apprently presents challenges for higher order algorithms.
- 10halo - contains 10 speech signals that are highly correlated (all 10 speakers say the same sentence).
- GnBand - contains 5 fourth order colored sources with a distribution close to Gaussian. This is a rather "difficult" benchmark.
- acspeech16 - contains 16 typical speech signals which have a temporal structure but are not precisely independent
- ABio5 - contains 5 typical biological sources
- ACsparse10 - contains 10 sparse (smooth bell-shape) sources that are approximately independent. The SOS blind source separation algorithms fail to separate such sources.
- 25SpeakersHALO - 25 highly correlated speech signals
- Vsparserand10 - very sparse random signals
- ACsincpos10 - positive sparse signals
- X5smooth - smooth signals
- Speech20 - 20 speech/music sources
- X10randsparse - random sparse signals
- 64soundsstd - a variety of sound sources (64 in total)

14 Details on Optimization Algorithm

Our optimization algorithm solves the general problem:

$$\min_x f(x) \tag{51}$$
$$\text{s.t. } c_i(x) = 0, \quad i = 1, 2, \ldots, m \tag{52}$$
$$\text{s.t. } g_j(x) \geq 0, \quad j = 1, 2, \ldots L \tag{53}$$

where $x \in R^n$.

We convert the inequality constraints into equality constraints via slack variables as follows:

$$g_j(x) - s_j = 0 \tag{54}$$
$$s_j \geq 0, \quad j = 1, 2, \ldots L \tag{55}$$

Thus the optimization problem becomes:

$$\min f(x) \tag{56}$$
$$\text{s.t. } c_i(x) = 0, \quad i = 1, 2, \ldots, m \tag{57}$$
$$\text{s.t. } g_j(x) - s_j = 0, \quad j = 1, 2, \ldots L \tag{58}$$
$$s_j \geq 0 \tag{59}$$

This problem is now an equality constrained problem where the inequalities have been replaced by the bound constraints on the slack variables. Thus it suffices to consider equality constrained problems with bounds on independent variables as follows:

$$\min f(x) \tag{60}$$
$$\text{s.t. } c_i(x) = 0, \quad i = 1, 2, \ldots, m \tag{61}$$
$$\text{s.t. } l_i \leq x_i \leq u_i, \quad i = 1, 2, \ldots n \tag{62}$$

where $x \in R^n$.

Our code uses a trust region based augmented lagrangian approach to solve these bound constrained problems following closely the LANCELOT software package [12], [10]. The augmented lagrangian function for the above problem is defined as:

$$\mathcal{L}(x, \lambda, \mu) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i=1}^{m} c_i(x)^2 \tag{63}$$

At each outer iteration k, given current values of λ^k and μ_k we solve the subproblem:

$$\min \mathcal{L}(x, \lambda^k, \mu_k) \tag{64}$$
$$\text{s.t. } l_i \leq x_i \leq u_i \tag{65}$$

If P is the projection operator defined as

$$[P(z,l,u)]_i = \begin{cases} l_i & \text{if} \quad z_i \leq l_i \\ z_i & \text{if} \quad l_i \leq z_i \leq u_i \\ u_i & \text{if} \quad z_i \geq u_i \end{cases} \tag{66}$$

then the Karush-Kuhn-Tucker (KKT) optimality condition for 64 is given as [10]:

$$x - P(x - \nabla_x \mathcal{L}(x, \lambda^k, \mu_k), l, u) = 0 \tag{67}$$

The outer iteration code is given in Framework 1. Note that the penalty parameter μ_k is updated based on a feasibility monitoring strategy that allows for a decrease in μ_k if sufficient accuracy is not achieved in solving the subproblem 64.

At each inner iteration we form a quadratic approximation to the augmented lagrangian and approximately solve the inequality constrained quadratic sub-problem:

$$\min_p \ \frac{1}{2} p^T \nabla^2_{xx} \mathcal{L}(x, \lambda, \mu) p + \nabla_x \mathcal{L}(x, \lambda, \mu)^T p \tag{68}$$
$$\text{s.t. } l_i \leq x_i \leq u_i \tag{69}$$
$$\text{s.t. } ||p||_\infty \leq \Delta \tag{70}$$

The inner iteration code uses non-linear gradient projection [5] followed by Newton-CG-Steihaug conjugate gradient iterations [36]. Quasi-Newton updates are performed using either SR1 [11] (recommended for non-convex functions) or BFGS [4] (recommended for convex functions). For very large problems, we switch to the limited memory variants [32] of these quasi-Newton approximations. The algorithm details are given in Framework 2. The trust region update code is based on a standard progress monitoring strategy [33] and is given in Framework 3.

15 Optimization Benchmarks

The optimization core of ADIS has been tested on many benchmark problems from the GAMS library at `http://www.gamsworld.org/performance` as well as benchmarks from MINOS [31].

Algorithm 1 F1: Outer Iteration

Require: Initial point x_{init}, λ^0, μ_0, $\theta^h \in (1,\infty)$, $\theta_l \in (0,1)$

Choose tolerances η^*_{con} and η^*_{grad}. The default in ADIS is $\eta^*_{con} = \eta^*_{grad} = 1e{-}6$. .
$\mu = \mu_0$, $\eta_{con} = 1/\mu_0^{0.1}$, $\eta_{grad} = 1/\mu_0$
for $k = 0, 1, 2, \ldots$ **do**
  $found = 0$
  while $found \neq 1$ **do**
    Try to find x_k such that
    $||x_k - P(x_k - \nabla_x \mathcal{L}(x_k, \lambda^k, \mu_k), l, u)||_\infty \leq \eta_{grad}$ via F2 using starting point as x_{k-1}.
    if above step is completed successfully **then**
      Set $found = 1$
    else
      $\lambda^{k+1} = \lambda_k$
      $\mu_{k+1} = \theta_l \mu_k$
      $\eta_{con} = 1/\mu_k^{0.1}$
      $\eta_{grad} = 1/\mu_k$
    end if
  end while
  if $||c(x_k)||_\infty \leq \eta_{con}$ **then**
    if $||c(x_k)||_\infty \leq \eta^*_{con}$ and
    $||x_k - P(x_k - \nabla_x \mathcal{L}(x_k, \lambda^k, 0), l, u)||_\infty \leq \eta^*_{grad}$ **then**
      Stop and return current solution x_k.
    end if
    $\lambda^{k+1} = \lambda_k - \mu_k c(x_k)$
    $\mu_{k+1} = \mu_k$
    $\eta_{con} = \eta_{con}/\mu_{k+1}^{0.9}$
    $\eta_{grad} = \eta_{grad}/\mu_{k+1}$
  else
    $\lambda^{k+1} = \lambda_k$
    $\mu_{k+1} = \theta_h \mu_k$
    $\eta_{con} = 1/\mu_k^{0.1}$
    $\eta_{grad} = 1/\mu_k$
  end if
end for

Algorithm 2 F2: Inner Iteration

Require: j_{max}, η_{grad}, Δ, l, u, λ^k, μ_k, $\eta \in (0,1)$, $flag$

$found = 0$

$x = x_{k-1}$, $j = 1$

Compute, $g = \nabla_x \mathcal{L}(x, \lambda^k, \mu_k)$

Estimate $B = \nabla^2_{xx} \mathcal{L}(x, \lambda^k, \mu_k)$ using BFGS, SR1 or limited memory BFGS, limited memory SR1 quasi Newton Updates.

while $found \neq 1$ and $j \leq j_{max}$ **do**

Calculate the Cauchy point p_c for problem:

$$\min_p \ \frac{1}{2} p^T B p + g^T p \tag{71}$$

$$\text{s.t. } l - x \leq p \leq u - x \tag{72}$$

$$\text{s.t. } ||p||_\infty \leq \Delta \tag{73}$$

using non-linear gradient projection and calculate the current active set $\mathcal{A}$. Let e_i be the unit vector with 1 at position i and zeros elsewhere. If $i_1, i_2, \ldots i_q \notin \mathcal{A}$ then let $\tilde{Q} = [e_{i_1}, e_{i_2}, \ldots, e_{i_q}]$.

$\tilde{g} = \tilde{Q}^T (g + B\, p_c)$ and $\tilde{B} = \tilde{Q}^T B \tilde{Q}$

Compute the approximate solution $\hat{v}$ to the problem

$$\min_v \ \frac{1}{2} v^T \tilde{B} v + \tilde{g}^T v \tag{74}$$

$$\text{s.t. } l - x \leq p_c + \tilde{Q} v \leq u - x \tag{75}$$

$$\text{s.t. } ||p_c + \tilde{Q} v||_\infty \leq \Delta \tag{76}$$

using truncated conjugate gradient iteration (Newton-CG, Steihaug). If $flag = 1$ use pre-conditioned Newton-CG using the inexact-modified Cholesky factorization.

Compute $\hat{p} = p_c + \tilde{Q}\hat{v}$

Calculate $\delta_{\mathcal{L}} = \mathcal{L}(x) - \mathcal{L}(x + \hat{p})$, $\delta_m = 0.5 \hat{p}^T B \hat{p} + g^T \hat{p}$ and $\rho = \delta_{\mathcal{L}} / \delta_m$

if $\rho > \eta$ **then**

$x = x + \hat{p}$

end if

Compute new trust region radius Δ using Framework F3.

Compute, $g = \nabla_x \mathcal{L}(x, \lambda^k, \mu_k)$ if $\rho > \eta$ holds otherwise use the previous value.

Estimate $B = \nabla^2_{xx} \mathcal{L}(x, \lambda^k, \mu_k)$ using BFGS, SR1 or limited memory BFGS, limited memory SR1 quasi Newton Updates. Do the update even if $\rho < \eta$.

if $||x - P(x - \nabla_x \mathcal{L}(x, \lambda^k, \mu_k), l, u)||_\infty \leq \eta_{grad}$ **then**

$found = 1$

end if

$j = j + 1$

end while

Algorithm 3 F3:Trust Region Update

Require: ρ, $\hat{p}$, Δ
if $\rho > 0.75$ **then**
  if $||\hat{p}||_\infty \leq 0.8\Delta$ **then**
    $\Delta = \Delta$
  else
    $\Delta = 2\Delta$
  end if
end if
if $0.1 \leq \rho \leq 0.75$ **then**
  $\Delta = \Delta$
else
  $\Delta = 0.5\Delta$
end if
return Δ

This section will describe some numerical experiments on interesting and difficult optimization benchmarks used to test the optimization core of ADIS. For these benchmarks, the gradient information was generated using automatic differentiation from the software package INTLAB [34]. A limited memory variant of symmetric rank 1 (SR1) updating was used. The CG iterations were not preconditioned. The convergence tolerances η_{con} and η_{tol} were set to their default value of $1e{-}6$.

Electron 50

Given n_p electrons, find the equilibrium state distribution (of minimal Columb potential) of the electrons positioned on a conducting sphere. This problem is from COPS3 [15] benchmark dataset.

The problem is to find a configuration of low energy for a given set of point charges on a conducting sphere. It originated with Thomson's plum pudding model of the atomic nucleus and is representative of an important class of problems in physics and chemistry that determine a structure with respect to atomic positions. Mathematically, the problem is:

$$\min_{x,y,z} \quad f(x,y,z) = \sum_{i=1}^{n_p-1} \sum_{j=i+1}^{n_p} [(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]^{-0.5} \tag{77}$$

subject to

$$x_i^2 + y_i^2 + z_i^2 = 1, i = 1, \ldots, n_p \tag{78}$$

The 150 variable problem for $n_p = 50$ was taken from GAMS performance library (PrincetonLib (NLP)) at `http://www.gamsworld.org/performance/princetonlib/htm/fekete/fekete2.htm`. The best known objective for this problem for $n_p = 50$ is $f^* = 1055.1823$. Our code attains this best objective in 13 outer iterations. See figure 13 for convergence diagnostics.

Non-negative Least Square (NNLS)

In NNLS we solve the problem:

$$\min_x \quad f = ||Ax - b||_2 \tag{79}$$

subject to:

$$Cx - d \geq 0 \tag{80}$$

We solve the 300 variable optimization problem taken from GAMS performance library (PrincetonLib (NLP)) at `http://www.gamsworld.org/performance/princetonlib/htm/nnls/nnls.htm`.

The best known object for this problem is $f^* = 633785.4462$. Our code attains this best objective in 26 outer iterations. See figure 14 for convergence diagnostics.

Largest Small PolyGon

This is a classic problem also from COPS3 [15] benchmark dataset. Given coordinates (r_i, θ_i) of the n_v vertices of a polygon, we wish to solve the problem:

$$\max_{r,\theta} \quad f(r, \theta) = 0.5 \sum_{i=1}^{n_v - 1} r_{i+1} r_i \sin(\theta_{i+1} - \theta_i) \tag{81}$$

subject to:

$$r_i^2 + r_j^2 - 2\, r_i r_j \cos(\theta_i - \theta_j) \leq 1,\ 1 \leq i < n_v,\ i < j \leq n_v \tag{82}$$

$$\theta_i \leq \theta_{i+1},\ 1 \leq i < n_v \tag{83}$$

$$\theta_i \in [0, \pi], r_i \geq 0,\ 1 \leq i \leq n_v \tag{84}$$

Some of the interesting features of this problem that make it difficult include the presence of the order of n_v^2 nonlinear nonconvex inequality constraints and the presence of $O(n_v!)$ local minima. See [15] for more details. We solve the $n_v = 6$ problem taken from GAMS performance library (PrincetonLib (NLP)) at `http://www.gamsworld.org/performance/princetonlib/htm/polygon/pgon.htm`.

The best known objective in GAMS library for this problem is $f^* = 0.5$.

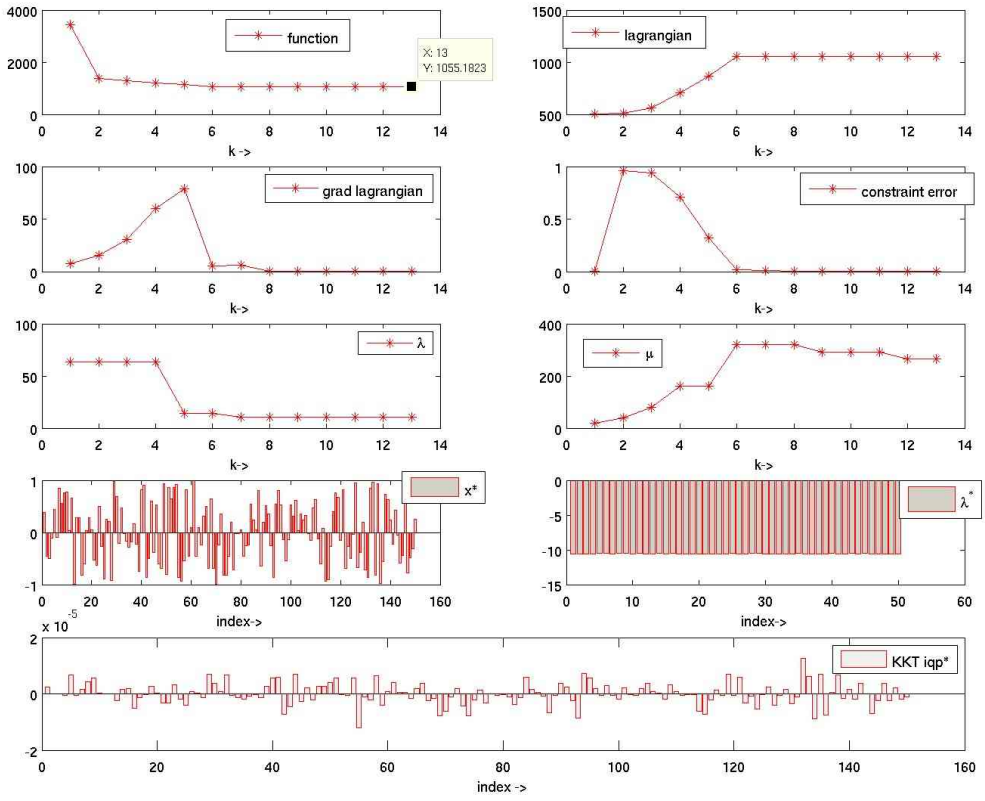


Figure 13: Electron 50 convergence diagnostics. The best objective of $f^* = 1055.1823$ was attained in 13 outer iterations. Figure shows the evolution of objective function, the Lagrangian, norm of the gradient of the Lagrangian, norm of the constraint satisfaction error, norm of the Lagrange multipliers and penalty parameter over algorithm iterations along with verification of KKT optimality conditions.

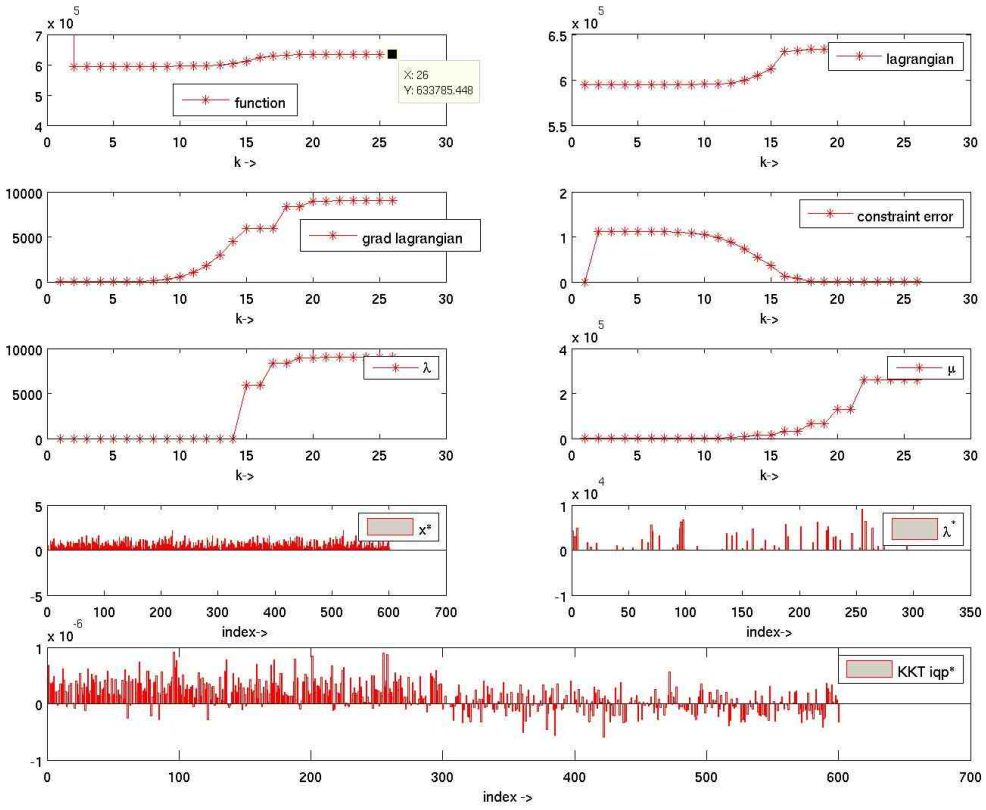


Figure 14: Non-negative least squares convergence diagnostics. The best objective of $f^* = 633785.44$ was attained in 26 outer iterations. Figure shows the evolution of objective function, the Lagrangian, norm of the gradient of the Lagrangian, norm of the constraint satisfaction error, norm of the Lagrange multipliers and penalty parameter over algorithm iterations along with verification of KKT optimality conditions.

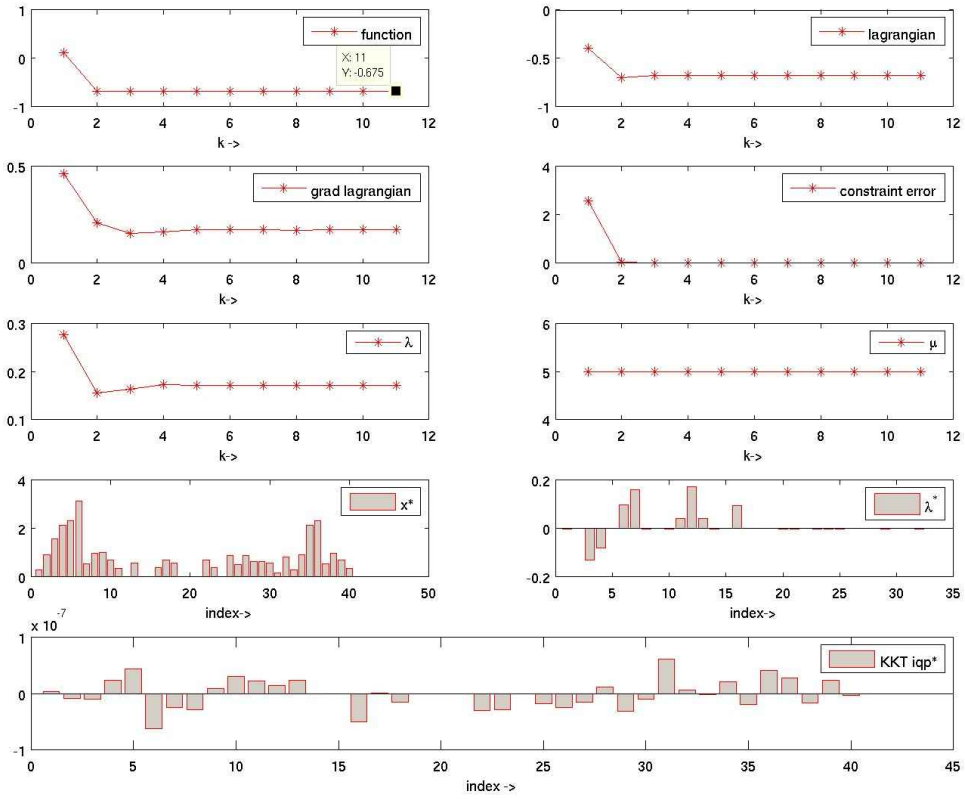


Figure 15: Largest Small Polygon convergence diagnostics for $n_v = 6$. Note that we first convert the problem into a minimization problem by multiplying the objective function with -1. The optimal objective value of -0.675 for the minimization problem was attained in 11 outer iterations. Figure shows the evolution of objective function, the Lagrangian, norm of the gradient of the Lagrangian, norm of the constraint satisfaction error, norm of the Lagrange multipliers and penalty parameter over algorithm iterations along with verification of KKT optimality conditions.

We first convert the maximization problem 81 into a minimization problem by multiplying the objective function by -1. We solve this problem and then evaluate the original objective 81 at the solution. We find that our algorithm attains an objective of $\hat{f} = 0.675$ in 11 outer iterations.

This is better than that reported by GAMS solvers. We were a little surprised by this observation. On researching this problem further we found that Graham [19] had solved this problem in 1975 and the best solution is indeed 0.675 (see `http://mathworld.wolfram.com/GrahamsBiggestLittleHexagon.html`). On plotting the optimal hexagon estimated by our code alongside the solution from [19], we find that they are identical. See figure 15 for convergence diagnostics.

References

[1] S. Amari, I. Cichocki, and H. Yang. A new learning algorithm for blind source separation. *Advances in Neural Information Processing Systems*, 8:757–763, 1996.

[2] C.F. Beckmann and S.M. Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans. on Medical Imaging*, 23(2):137–152, 2004.

[3] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[4] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *J. Ins. Math. Applcs.*, 6:76–90, 1970.

[5] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ci You Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(6):1190–1208, 1995.

[6] J. F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagolization. *SIAM Journal of Matrix Analysis and Applications*, 17(1):161–164, 1996.

[7] A. Cichocki, S. Amari, K. Siwek, T. Tanaka, Anh Huy Phan, and R. Zdunek. ICALAB - MATLAB Toolboxes Ver. 3 for signal processing. Technical report, Laboratory for Advanced Brain Signal Processing, Japan, http://www.bsp.brain.riken.jp/ICALAB, March 2007.

[8] Andrzej Cichocki and Shun-ichi Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. John Wiley, Chichester, UK, 2003.

[9] P. Comon. 'Independent Component Analysis, a new concept ? *Signal Processing*, 36(3):287–314, 1994.

[10] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numerical Analysis*, 28:545–572, 1991.

[11] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Convergence of quasi-Newton matrices generated by the Symmetric Rank One update. *Math. Programming*, 50:177–196, 1991.

[12] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT: A Fortan package for large-scale nonlinear optimization (Release A). *Springer Series in Computational Mathematics*, 17, 1992.

[13] S. Cruces, L. Castedo, and L. Cichocki. Robust blind source separation algorithms using cumulants. *Neurocomputing*, 49:87–117, 2002.

[14] S. Cruces, A. Cichocki, and L Castedo. Blind source extraction in gaussian noise. volume 1, pages 63–68, Helsinki, Finland, June 2000. Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Signal Separation (ICA'2000).

[15] Elizabeth D. Dolan, Jorge J. More, and Todd S. Munson. Benchmarking Optimization Software with COPS3. Technical Report ANL/MCS-TM-273, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, February 2004.

[16] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.

[17] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. of Computers*, 23(9):881–890, 1974.

[18] Xavier Giannakapoulos, Juha Karhunen, and Erkki Oja. An Experimental Comparison of Neural ICA Algorithms. pages 651–656. In Proc. Int. Conf. on Artificial Neural Networks, (ICANN'98), 1998.

[19] R. L. Graham. The Largest Small Hexagon. *J. Combin. Th. Ser. A*, 18:165–170, 1975.

[20] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

[21] J. Himberg, A. Hyvarinen, and F. Esposito. Validating the independent components of neuroimaging time-series via clustering and visualization. *NeuroImage*, 22(3):1214–1222, 2004.

[22] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[23] A. Hyvarinen. New approximations of differential entropy for independent component analysis and projection pursuit. *Advances in Neural Information Processing Systems*, 10:273–279, 1998.

[24] A. Hyvarinen. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[25] A. Hyvarinen. Survey on Independent Component Analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[26] A. Hyvarinen and E. Oja. Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13(4-5):411–430, 2000.

[27] Z. Koldovsky, P. Tichavsky, and E. Oja. Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramer-Rao Lowr Bound. *IEEE Trans. on Neural Networks*, 17(5):1265–1277, 2006.

[28] Z. Koldovsky, P. Tichavsky, and E. Oja. Performance Analysis of the FastICA Algorithm and Cramer-Rao Bounds for Linear Independent Component Analysis. *IEEE Trans. on Signal Processing*, 54(4), 2006.

[29] T. Minka. Automatic choice of dimensionality for PCA. Technical Report Technical Report 514, MIT, Cambridge, 2000.

[30] J. More and D. Sorensen. Computing a trust region step. *SIAM J. Sci. Stat. Comp.*, 4:533–572, 1983.

[31] B. A. Murtagh and M. A. Saunders. MINOS 5.4 User's Guide. Technical Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA 94305, 1993.

[32] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Computation*, 35:773–782, 1980.

[33] J. Nocedal and S.J Wright. *Numerical Optimization, 2nd Edition*. New York:Springer, 2006.

[34] S. M. Rump. INTLAB: Interval laboratory, a MATLAB toolbox for interval arithmetic.

[35] S.M. Smith, M. Jenkinson, M.W. Woolrich, C.F. Beckmann, T.E.J. Behrens, H. Johansen-Berg, P.R. Bannister, Luca M. De, I. Drobnjak, D.E. Flitney, R. Niazy, J. Saunders, J. Vickers, Y. Zhang, N. De Stefano, J.M. Brady, and P.M. Matthews. Advances in funcional and structural MR image analysis and implementation as FSL. *NeuroImage*, 23(S1):208–219, 2004.

[36] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numerical Analysis*, 20:626–637, 1983.

[37] Emmanuel Vincent, Remi Gribonval, and Cedric Fevotte. Performance Measurement in Blind Audio Source Separation. *IEEE Trans. on Audio, Speech and Language Processing*, 14(4), 2006.

[38] M. W. Woolrich, B. D. Ripley, J. M. Brady, and S. M. Smith. Temporal Autocorrelation in univariate linear modeling of FMRI data. *NeuroImage*, 14(6):1370–1386, 2001.

[39] Vicente Zarzoso and Pierre Comon. "How Fast is FastICA". "14th European Signal Processing Conference (EUSIPCO), Firenze, Italy.", September 2006.